

Grado en Ingeniería Informática  
2018/2019

*Trabajo Fin de Grado*

# “Sistema de Audio Compartido entre Dispositivos Móviles”

---

José Julio Corps Orgaz

Tutor: Ángel García Crespo  
Leganés, septiembre de 2019



Esta obra se encuentra sujeta a la licencia Creative Commons  
**Reconocimiento – No Comercial – Sin Obra Derivada**



## RESUMEN

El objetivo de este proyecto es implementar una aplicación para teléfonos móviles con sistema operativo Android, que permita enviar audio de unos dispositivos a otros sin necesidad de tener conexión a Internet. Para ello, se establece una ‘conferencia’ donde están presentes una serie de teléfonos móviles. El audio se graba por el micrófono de los dispositivos y se envía y reproduce en el resto de dispositivos participantes en la conferencia.

Para conseguir el objetivo, una vez estudiadas las diferentes alternativas posibles (comunicar los dispositivos mediante Wi-Fi Direct, Bluetooth o una red P2P), se llega a la conclusión que la tecnología que mejor cumple el objetivo es Wi-Fi Direct, por tanto, es la que se usa para implementar la aplicación.

**Palabras clave:** Android, Wi-Fi Direct, Java, audio, comunicación, dispositivo móvil, conferencia



## ABSTRACT

The goal for this project is to implement a mobile application for Android devices that allows sending audio between devices without an active Internet connection. To reach this goal, a 'conference' (that is, a group of mobile devices) is created. Input audio is recorded through the microphone in each device, then sent real-time to every other device in the conference, and finally played through their default audio output system.

Through a thorough candidate-technology analysis (including Wi-Fi Direct, Bluetooth and P2P networks as communication systems), the most suitable technology found to meet the goal is Wi-Fi Direct. Thus, this is the technology used to implement the application.

**Keywords:** Android, Wi-Fi Direct, Java, audio, communication, mobile device, conference



## **AGRADECIMIENTOS**

A mi familia por haberme ayudado tantísimo en estos años de carrera. Sin ellos, no habría sido posible estar donde estoy hoy.

También agradecer a mi tutor, Ángel, que no ha sido sólo mi tutor, si no la primera persona que me ha dado una oportunidad laboral en SoftLab. Gracias también a mis compañeros de SofLab, por todo lo que me han aportado.





# I. Índice general

<b>RESUMEN</b>	<b>III</b>
<b>ABSTRACT</b>	<b>V</b>
<b>AGRADECIMIENTOS</b>	<b>VII</b>
<b>1 CAPÍTULO 1 – INTRODUCCIÓN</b>	<b>2</b>
1.1 Visión general (Ámbito y Alcance)	2
1.2 Motivación (Contexto)	2
1.3 Objetivos	3
1.4 Estructura del documento	4
<b>2 CAPÍTULO 2 – ESTADO DEL ARTE</b>	<b>6</b>
2.1 Explicación de las tecnologías involucradas	6
2.1.1 Android	6
2.1.2 Wi-Fi Direct	8
2.1.3 Bluetooth	10
2.1.4 Arquitectura cliente-servidor	11
2.1.5 Arquitectura P2P ( <i>peer-to-peer</i> )	12
2.2 Aplicaciones disponibles en el mercado	13
2.2.1 Skype	13
2.2.2 Hangouts	13
2.2.3 Facetime	14
2.2.4 Discord	14
<b>3 CAPÍTULO 3 – ANÁLISIS DE LA SOLUCIÓN</b>	<b>16</b>
3.1 Selección de la solución	16
3.1.1 Elección del tipo de aplicación a implementar	16
3.1.1.1 Aplicación nativa	16
3.1.1.2 Aplicación web	17
3.1.1.3 Aplicación híbrida	18
3.1.2 Elección del lenguaje de programación	19
3.1.3 Elección de la tecnología a usar para comunicar los dispositivos	20
3.1.4 Elección del protocolo a usar para comunicar los dispositivos	21
3.2 Comparativa de la solución con trabajos similares	21
<b>4 CAPÍTULO 4 – DISEÑO E IMPLEMENTACIÓN DEL SISTEMA</b>	<b>24</b>
4.1 Fases	24
4.1.1 Fase de análisis de compatibilidad y estado	24
4.1.2 Fase de exploración	25
4.1.3 Fase de conexión	28
4.1.4 Fase de comunicación	30
4.1.5 Fase de desconexión	34

4.2	Resumen de funcionamiento .....	35
4.3	Captura de sonido .....	36
4.4	Reproducción del audio .....	36
5	<b>CAPÍTULO 5 – PLANIFICACIÓN .....</b>	<b>38</b>
5.1	Planificación inicial .....	38
5.2	Duración real del proyecto .....	39
6	<b>CAPÍTULO 6 – ENTORNO SOCIO-ECONÓMICO.....</b>	<b>42</b>
6.1	Presupuesto .....	42
6.1.1	Gastos directos .....	42
6.1.1.1	Coste del personal que trabaja en el proyecto .....	42
6.1.1.2	Coste del software .....	43
6.1.1.3	Coste de los aparatos electrónicos e informáticos.....	45
6.1.2	Gastos indirectos y porcentaje de beneficio.....	45
6.1.2.1	Coste total .....	46
6.2	Impacto socio-económico.....	47
6.2.1	Análisis del panorama actual .....	47
6.2.2	Análisis del impacto social .....	48
6.2.3	Análisis del impacto económico .....	49
7	<b>CAPÍTULO 7 – MARCO REGULADOR .....</b>	<b>50</b>
7.1	Privacidad y protección de datos .....	50
7.2	Procesado del audio.....	50
7.3	Transporte del audio .....	51
7.4	Propiedad intelectual .....	51
7.5	Memoria escrita.....	52
7.6	Código fuente .....	52
7.7	Dependencias .....	52
8	<b>CAPÍTULO 8 – CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS...</b>	<b>54</b>
8.1	Conclusiones .....	54
8.2	Líneas de trabajo futuras .....	55
	<b>BIBLIOGRAFÍA .....</b>	<b>56</b>
	<b>ANEXO I: RESUMEN EN INGLÉS .....</b>	<b>60</b>
	<b>AI.1 Abstract.....</b>	<b>60</b>
	<b>AI.2 Introduction.....</b>	<b>60</b>
	AI.2.1 Overview (Area and Scope) .....	60
	AI.2.2 Motivation (Context).....	61
	<b>AI.3 Goals.....</b>	<b>62</b>
	<b>AI.4 Solution analysis.....</b>	<b>63</b>
	AI.4.1 Selection of a type of application to implement.....	63
	AI.4.2 Selection of a programming language .....	64
	AI.4.3 Selection of a technology for communicating devices.....	64
	AI.4.4 Selection of the protocol for device communications .....	64
	AI.4.5 Comparision of the selected solution and its alternatives .....	65

<b>AI.5 Design and implementation .....</b>	<b>66</b>
<b>AI.5.1 Summary of operation.....</b>	<b>67</b>
<b>AI.5.2 Audio Capture and playback.....</b>	<b>68</b>
<b>AI.5 Conclusion .....</b>	<b>69</b>
<b>AI.5 Future lines of work.....</b>	<b>70</b>



## II. Índice de Figuras

Fig 2.1 Comparativa entre un esquema con punto de acceso (izquierda) y un esquema Wi-Fi Direct (derecha).....	8
Fig 2.2 Comparativa entre un esquema cliente/sevidor (izquierda) y un esquema P2P (derecha) .....	12
Fig 4.1: Descripción de la fase de exploración .....	26
Fig 4.2: Pantalla de exploración de los dispositivos disponibles .....	27
Fig 4.3: Selección de un dispositivo para la conexión .....	29
Fig 4.4: Comunicación entre dos dispositivos .....	31
Fig 4.5: Comunicación entre N dispositivos.....	32
Fig 4.6: Pantalla de conferencia .....	33
Fig 4.7: Grafo de transición entre fases .....	35
Fig A.1 Stage graph .....	67



### III. Índice de Tablas

TABLA 2.1: VENTAS MUNDIALES DE TELÉFONOS MÓVILES POR SISTEMA OPERATIVO EN EL AÑO 2017 [2] .....	7
TABLA 3.1: COMPARATIVA ENTRE ALGUNAS APLICACIONES DISPONIBLES EN EL MERCADO Y LA IMPLEMENTADA EN ESTE PROYECTO .....	21
TABLA 5.1: DIAGRAMA DE GANTT DE LA PLANIFICACIÓN INICIAL DEL PROYECTO .....	39
TABLA 5.2: DURACIÓN ESTIMADA DE CADA SPRINT EN SEMANAS .....	39
TABLA 5.3: DURACIÓN REAL DE CADA SPRINT EN SEMANAS.....	40
TABLA 6.1: COSTE DEL PERSONAL QUE TRABAJA EN EL PROYECTO .....	43
TABLA 6.2: COSTE DEL SOFTWARE .....	44
TABLA 6.3: COSTE DE LOS APARATOS ELECTRÓNICOS E INFORMÁTICOS	45
TABLA 6.4: COSTES INDIRECTOS ASOCIADOS AL PROYECTO .....	46
TABLA 6.5: COSTE TOTAL DEL PROYECTO (SIN IVA) .....	46
TABLA A.1: COMPARISON BETWEEN SOME AVAILABLE APPLICATIONS ON THE MARKET AND THE SELECTED SOLUTION .....	65





# **1 CAPÍTULO 1 – INTRODUCCIÓN**

En este capítulo se realiza una breve presentación de este proyecto. Se explica el ámbito y el alcance, se expone la motivación del proyecto y se enumeran los diferentes objetivos que se quieren lograr en este Trabajo de Fin de Grado. Finalmente se explica cuál es la estructura del presente documento.

## **1.1 Visión general (Ámbito y Alcance)**

Este proyecto se ha realizado como Trabajo de Fin de Grado del Grado en Ingeniería Informática. Por tanto, su ámbito es el de la ingeniería informática.

Lo que se quiere conseguir en este proyecto es realizar un Sistema de Audio Compartido entre dispositivos móviles. Dicho sistema envía el audio recogido por el micrófono de los teléfonos móviles a otros teléfonos móviles. El proyecto, por tanto, requiere conocimientos en el ámbito de las redes de comunicaciones y de la teoría de información, ya que hay que diseñar e implementar una aplicación capaz de procesar, transmitir y recibir información, que estará representada en forma de audio.

Existen, principalmente dos sistemas operativos móviles: iOS y Android [1]. Para conseguir el objetivo de este proyecto, se ha realizado una aplicación móvil para que sea soportada por móviles cuyo sistema operativo sea Android, no se ha realizado una versión para iOS.

El alcance de este proyecto, aparte de la implementación, es la realización de este documento, en el que se incluye un análisis de la solución, un diseño técnico y un presupuesto del proyecto.

## **1.2 Motivación (Contexto)**

Con el avance de la tecnología, la forma de comunicarse ha cambiado drásticamente. Hemos pasado de comunicarnos solo por carta y llamadas telefónicas, a hacerlo por mensajería instantánea, vídeo llamadas, llamadas por internet, etc.

La aplicación implementada en este proyecto sirve para comunicarse, pero sin necesidad de ningún tipo de conexión a ningún servidor, la conexión se realiza directamente hacia los dispositivos con los que se quiere establecer la comunicación. Esto es algo muy novedoso y útil, ya que podemos establecer una comunicación en cualquier lugar, siendo necesario únicamente poseer un dispositivo móvil. No es necesario nada más. Por eso, cambia la manera de comunicarse, ya que se elimina la necesidad de conexión a una red de datos móviles, conexión a internet o cobertura móvil. Sin embargo, esto tiene un inconveniente: realizar la conexión directamente a los dispositivos, tiene como requerimiento que estos se encuentren próximos entre sí, ya que, si no, los dispositivos no pueden comunicarse directamente entre sí. Cuenta también con múltiples ventajas, algunas de ellas son:

- Podemos comunicarnos (siempre y cuando el dispositivo al que nos queramos comunicar esté próximo) si estamos en un barco en mitad del océano, en un avión, en un sótano, etc.
- Se mejora la privacidad, ya que, al no pasar los datos por internet, la probabilidad de acceso y robo de información es más baja.

En este contexto, es en el que tiene cabida la realización del proyecto.

### **1.3 Objetivos**

El objetivo principal que se quiere conseguir con este Trabajo de Fin de Grado es conectar dos dispositivos móviles para que puedan compartir audio entre ellos.

Compartir audio es algo muy genérico, puede haber diferentes formas:

- Compartir una pista de audio ya grabada (canciones, notas de voz, etc.).
- Compartir audio en tiempo real de manera bidireccional (llamada telefónica, llamada por Skype, etc.).
- Compartir audio en tiempo real de manera unidireccional (un dispositivo actúa como emisor de audio y otros como receptores), como por ejemplo un Walkie Talkie, donde mientras un dispositivo manda audio, no puede recibir el audio

entrante de los demás, que tendrían que esperar a que el dispositivo deje de emitir para poder emitir ellos y ser recibidos.

Este proyecto tiene como finalidad conseguir compartir audio de la segunda forma: en tiempo real de manera bidireccional. Hay muchas aplicaciones ya existentes que realizan esta tarea, pero la novedad que se quiere aportar con este proyecto es eliminar el servidor que hay entre los dispositivos en el resto de aplicaciones. Por tanto, se elimina la necesidad de conexión a Internet u a otra red.

El objetivo puede parecer muy general, por lo que se puede descomponer en diferentes sub-objetivos:

- Debe funcionar en móviles con sistema operativo Android.
- La aplicación debe funcionar con al menos 2 móviles conectados.
- La conexión entre los dispositivos se realiza sin necesidad de conexión a Internet.

En resumen: se quiere conseguir un sistema de audio compartido para dispositivos móviles que comparta audio de manera bidireccional en tiempo real, sin que sea necesario un servidor entre los dispositivos (por lo que no es necesaria la conexión a Internet).

#### **1.4 Estructura del documento**

El presente documento está estructurado de la siguiente manera:

- En el capítulo uno, se realiza un breve resumen del proyecto, para posteriormente realizar una pequeña introducción.
- En el capítulo dos, se realiza un planteamiento del problema y se detalla el estado del arte, donde se exponen las tecnologías involucradas en el proyecto y se realiza una comparación de la aplicación con otras que están ya disponibles en el mercado, realizando antes una breve descripción de estas.
- En el capítulo tres, se realiza un análisis de la solución, analizando las diferentes alternativas de solución y justificando el porqué de la elección de una de ellas.
- En el capítulo cuatro, se realiza el diseño técnico del sistema y se explica la implementación llevada a cabo.

- En el capítulo cinco, se realiza una planificación del proyecto, realizando una planificación inicial y explicando a continuación cual ha sido la duración real del proyecto.
- En el capítulo seis, se expone el entorno socio-económico, donde se detalla el presupuesto del proyecto y se analiza el impacto socio-económico.
- En el séptimo capítulo, se analiza el marco regulador del sistema, exponiendo la regulación aplicable al proyecto.
- En el capítulo ocho, se exponen las conclusiones de fin de proyecto y se realiza un breve análisis de las posibles líneas de trabajo futuras que puede tener el proyecto.
- Finalmente, se adjunta la bibliografía consultada para la realización del proyecto y como anexo, se añade un resumen en inglés del presente Trabajo de Fin de Grado.

## **2 CAPÍTULO 2 – ESTADO DEL ARTE**

En este capítulo, se realiza una descripción de las tecnologías involucradas en el proyecto y se describen algunas aplicaciones disponibles en el mercado cuya funcionalidad es similar a la de la aplicación implementada.

### **2.1 Explicación de las tecnologías involucradas**

En este capítulo se explica de manera general todas las tecnologías involucradas en este proyecto. Cabe destacar, que, de las explicadas en este apartado, no se han usado todas en la implementación, pero se han tenido que tener en cuenta y estudiar su viabilidad para llegar a la solución más óptima.

La aplicación móvil que da solución al problema planteado se ha desarrollado para teléfonos móviles con sistema operativo Android. Para conseguir la comunicación entre los dispositivos Android, se usa la tecnología Wi-fi Direct. Por lo tanto, en primer lugar, se describe Android y Wi-Fi direct.

Como se explica en el *Capítulo 3 – Análisis de la Solución*, para realizar la comunicación entre los dispositivos, se podrían usar otras tecnologías, como Bluetooth o P2P, por lo que en este capítulo también se explican dichas tecnologías.

#### **2.1.1 Android**

Android es un sistema operativo para dispositivos móviles, principalmente teléfonos inteligentes, aunque también está presente en televisiones, relojes, tabletas, etc. Existen más sistemas operativos para dispositivos móviles, aunque Android es el más usado, según se puede observar en la siguiente tabla:

TABLA 2.1: VENTAS MUNDIALES DE TELÉFONOS MÓVILES POR SISTEMA OPERATIVO EN EL AÑO 2017 [2]

Sistema Operativo	Unidades (en miles de unidades)	Porcentaje de unidades vendidas
Android	1.320.118,1	85,9%
iOS	214.924,4	14%
Otros Sistemas Operativos	1.493	0,1%

Fue lanzado en el año 2008 [3], y a lo largo de su vida ha tenido diferentes actualizaciones. Su versión actual es Android 10, la cual fue lanzada el 3 de septiembre de 2019 [4].

Al ser un sistema operativo, Android se encarga de gestionar el hardware, administrar recursos e interactuar con el usuario que esté usando el dispositivo móvil en cuestión.

Para realizar una aplicación que pueda ser ejecutada en un dispositivo cuyo sistema operativo sea Android, se ha usado hasta ahora, principalmente, Java como lenguaje de programación. Actualmente, hay otro lenguaje que está ganando mucha importancia, Kotlin. En realidad, se puede usar cualquier lenguaje de programación, siempre que la máquina virtual de Java del dispositivo (JVM) sea capaz de compilar y ejecutar ese lenguaje.

Hasta hace poco, Java y C++ eran los lenguajes oficiales de Android, pero en 2017 se añadió también Kotlin [5]. Una de las ventajas de Kotlin es que es fácil de aprender, además se conecta muy bien con Java, por lo que se pueden usar ambos en una misma aplicación.

C++ se usa cuando se necesita realizar parte de la aplicación de manera nativa. Para ello, disponemos de Android NDK (Android Native Development Kit). De esta manera, no es necesario que el código ejecute en la máquina virtual de Java del dispositivo, haciéndolo directamente sobre él. Así, se consigue un mayor rendimiento, ya que se elimina la máquina virtual.

Java, es el lenguaje que tradicionalmente se ha considerado nativo para Android, ya que tiene todas las bibliotecas y SDK de Android, aunque ejecute el código en una máquina virtual de Java. Kotlin funciona de la misma manera: contiene todas las bibliotecas y SDK

de Android y ejecuta en la máquina virtual de Java. La máquina virtual de Java de los dispositivos (JVM) realiza la conversión del código a instrucciones máquina de la plataforma en concreto en la que ejecute la JVM, para el dispositivo ejecute dichas instrucciones, en tiempo real. Por esta razón, si se quiere realizar una aplicación con una latencia muy baja es recomendable usar C++, ya que no necesita esta conversión.

La principal desventaja que tiene programar en C++ frente a Java, es la complejidad de programación. Muchas veces, se programan aplicaciones únicamente en Java, sin parte en C++ por la complejidad, sin tener en cuenta el rendimiento.

### 2.1.2 Wi-Fi Direct

Wi-Fi Direct es una tecnología que permite conectar directamente entre sí dos dispositivos, sin necesidad que haya algo entre ellos que los conecte. Al igual que los dispositivos (clientes) se conectan a un router (punto de acceso) para obtener conexión a internet, los dispositivos se pueden conectar entre sí mediante Wi-Fi Direct. Al conectarse dos o varios dispositivos por Wi-Fi Direct, uno de ellos será el punto de acceso, mientras que el resto serán clientes. Elegir cuál de ellos es el punto de acceso, se realiza mediante un acuerdo entre los dispositivos.

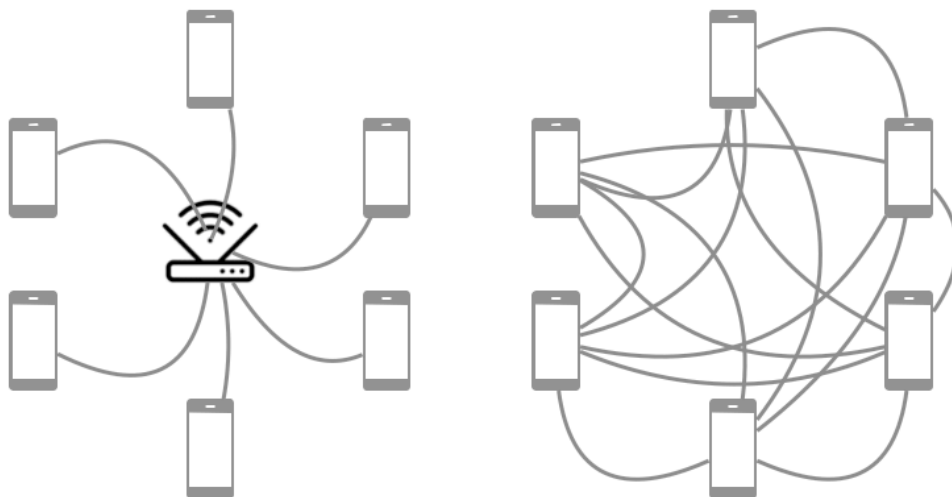


Fig 2.1 Comparativa entre un esquema con punto de acceso (izquierda) y un esquema Wi-Fi Direct (derecha)

Con esta tecnología, se pueden enviar datos de un dispositivo a otro, al igual que se podría hacer mediante Bluetooth o mediante conexión por cable. Sin embargo, mediante Wi-Fi Direct se obtienen una serie de ventajas:

1. No se necesita ningún tipo de cable, por lo que los dispositivos se pueden conectar con total libertad sin, sin el engorro de tener cables de por medio.
2. Tiene un rango de alcance mayor al de otros protocolos, como puede ser el Bluetooth. Su alcance es de 200 metros [6].
3. La velocidad de transmisión es más elevada que la de otros protocolos, como el Bluetooth.
4. No comprime el tamaño de los archivos que se comparten, como ocurre con algunas aplicaciones de mensajería instantánea que permiten compartir archivos como Whatsapp.
5. Permite compartir información entre dispositivos aunque estos se encuentren sin conexión a internet.

Entre sus principales desventajas encontramos:

1. Puede ser sensible a interferencias ya que se trata de una conexión inalámbrica.
2. No existe en los dispositivos una opción de fábrica.

Se puede decir que Wi-Fi Direct es una mejora del Bluetooth, sus aplicaciones son similares. Pero funciona de manera más rápida y segura. El protocolo de seguridad que utiliza es WPA2 [7].

Algunos ejemplos de aplicaciones del Wi-Fi Direct a situaciones que se pueden dar en la vida real son los siguientes:

1. Conectar dos móviles para poder compartir información entre ellos, como imágenes, vídeos, música, documentos, etc. Además, esta información se puede compartir con una mayor privacidad, ya que no es necesario dar información personal (como un número de teléfono) para conectarse con el otro dispositivo con el que se desea compartir información.



2. Conectar una impresora con otro dispositivo, para poder mandar documentos a impresión a la impresora, enviar de la impresora al dispositivo los documentos escaneados, etc.

Ejemplos de aplicaciones que ya están en el mercado y que usan Wi-Fi Direct son:

- **Feem:** Herramienta que permite compartir archivos entre los dispositivos que tengan esta aplicación instalada y se encuentren dentro de rango. Es válida para los sistemas operativos Android, iOS, macOS, Windows y Linux, permitiendo que en una misma transferencia haya dispositivos con cualquiera de estos SSOO.
- **FireChat:** Es una aplicación de mensajería instantánea que funciona sin necesidad de conexión a internet. Al no necesitar conectividad a la red, ha sido muy usada en algunas protestas en lugares en los que los gobiernos controlan y monitorizan la conexión a internet y el uso de redes sociales.

Como se explicará más adelante en este documento (*Capítulo 3 – Análisis de la Solución*), Wi-Fi Direct será la tecnología elegida para resolver la implementación de este Trabajo de Fin de Grado.

### 2.1.3 Bluetooth

El bluetooth es una especificación que permite conectar dispositivos que posean esta tecnología, mediante ondas electromagnéticas en la frecuencia de 2.4 Ghz [8]. Puede llegar a tener un alcance de hasta 100 metros de distancia [9].

La aparición del bluetooth fue una revolución, ya que permite conectar sin ningún tipo de cableado, dispositivos muy variados. Se trata pues, de una red inalámbrica, que está muy extendida en la actualidad, ya que hay infinidad de dispositivos electrónicos que la poseen, desde teléfonos móviles y ordenadores, pasando por teclados, ratones, auriculares, pantallas, televisiones, coches, etc. Que esté tan extendida, se explica por sus diversas ventajas:

- Su precio es muy bajo y es una tecnología que tiene una gran facilidad de implantación en dispositivos de pequeño tamaño.
- Compatibilidad entre diferentes dispositivos: sea cual sea el fabricante o sistema operativo de dicho dispositivo; se puede realizar una conexión sin ningún tipo de problema: por ejemplo, se puede conectar un teclado a un ordenador (ambos son dispositivos diferentes y pueden ser perfectamente de diferentes fabricantes), y el funcionamiento será el esperado.
- Por último, cabe destacar la facilidad y rapidez de conexión de un dispositivo con otro. Podemos conectar unos auriculares a un teléfono móvil pulsando simplemente un botón.

Aunque existen aplicaciones que usen bluetooth (por ejemplo, para mandar audio o texto, como las que se han explicado que usan Wi-Fi Direct), no se mencionan en este apartado, ya que el uso más extenso que se da al bluetooth es para conectar dispositivos, para lo que no hace falta ningún tipo de aplicación. Una vez los dispositivos estén conectados, pueden compartir información entre ellos.

#### **2.1.4 Arquitectura cliente-servidor**

Es una arquitectura que gira en torno a dos roles: cliente y servidor. El servidor se encarga de proporcionar servicios (normalmente expuestos a través de una red) y el cliente consume dichos servicios.

Este esquema, en su visión más tradicional, tiende a la centralización de recursos y servicios: el servidor realiza la mayor parte de tareas que requieren gran capacidad de cómputo y los clientes, aliviados de carga y responsabilidad, dependen estrechamente de él para poder funcionar. Es decir, los clientes no pueden realizar sus funciones ante la falta de servidor.

En la actualidad, existe mayor descentralización y los roles están más difuminados: los servidores actúan como clientes de otros servidores y los clientes también exponen funcionalidad de servidor. Un ejemplo de esto, son los microservicios, en los que las tareas que tendría un servidor monolítico se descomponen en pequeños servidores

encargados de tareas muy concretas. Estos pequeños servidores pueden tener interdependencias entre ellos.

### 2.1.5 Arquitectura P2P (*peer-to-peer*)

Es una arquitectura que se plantea frente a la arquitectura de cliente-servidor. El planteamiento cliente-servidor define dos roles claramente diferenciados. Por un lado, está el servidor, que ofrece servicios o realiza funciones complejas, y por otro el cliente, que consume dichos servicios o funcionalidades. Las arquitecturas P2P no establecen roles tan inflexibles permitiendo que un mismo elemento de la arquitectura pueda actuar como cliente o como servidor indistintamente, permitiendo la creación de redes colaborativas en las que existe un reparto de tareas más equilibrado para todos los equipos involucrados. Se produce, por tanto, una descentralización.

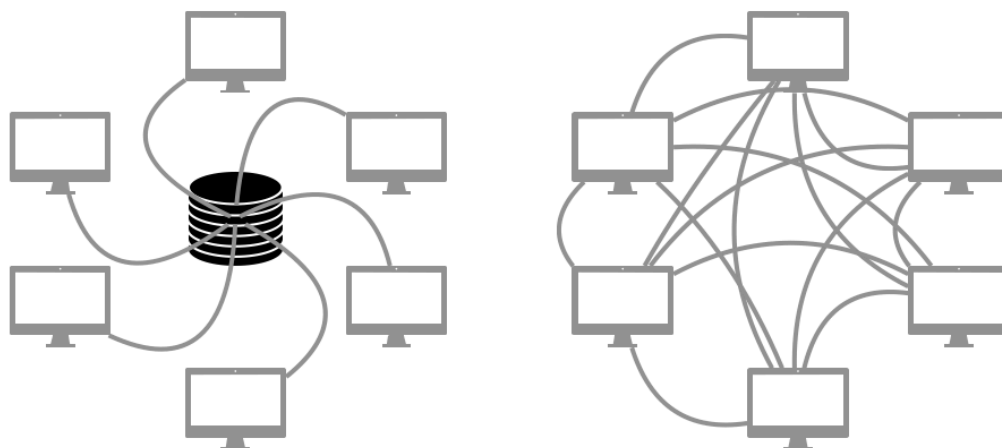


Fig 2.2 Comparativa entre un esquema cliente/servidor (izquierda) y un esquema P2P (derecha)

Algunas aplicaciones ya implementadas y disponibles en el mercado que usan P2P son:

- **uTorrent:** Es una aplicación de intercambio de contenido (música, vídeo, etc.). Para realizar dicho intercambio, usa el protocolo BitTorrent, basado en P2P.
- **Emule:** Al igual que uTorrent, también es una aplicación de intercambio de contenido, basada en P2P para realizar dicho intercambio.

## **2.2 Aplicaciones disponibles en el mercado**

### **2.2.1 Skype**

Es una aplicación multiplataforma (hay versión para iOS, Android, Windows, macOS, Linux y versión web) desarrollada por Microsoft. Permite comunicación mediante voz, vídeo y texto, pudiéndose desarrollar los tres tipos en una misma conversación. Esta comunicación se puede realizar siempre que haya conexión a internet, independientemente de si están conectados a través de una aplicación o desde el navegador web.

Es una aplicación gratuita, existiendo una cuota mensual mediante la cual se pueden realizar llamadas telefónicas y enviar mensajes de texto a números de teléfono, aunque no tengan Skype.

Toda la información que se comparte a través de Skype, se encuentra cifrada, pero si se hace una llamada telefónica convencional, el contenido no está cifrado. Si la información que se comparte son mensajes instantáneos, se usa el protocolo TLS (Transport Layer Security) y el algoritmo AES (Advanced Encryption Standard) [10].

### **2.2.2 Hangouts**

Es una aplicación desarrollada por Google, permite la comunicación mediante voz, vídeo y texto. La funcionalidad es similar a la de Skype. Hay una versión para iOS y Android. También se puede usar desde el navegador web. Los usuarios pueden entablar una conversación siempre que tengan conexión a internet, independientemente de si están conectados a través de una aplicación de un teléfono móvil o la versión web. La aplicación es totalmente gratuita, siendo el único requisito de uso que los usuarios tengan una cuenta de Google.

### **2.2.3 Facetime**

Es una aplicación desarrollada por Apple, que permite la comunicación por medio de voz y vídeo. Únicamente está disponible para dispositivos de Apple, hay versión para los sistemas operativos de iOS y macOS. No hay versión web. La aplicación es gratuita, teniendo como único requisito disponer un dispositivo Apple.

La principal diferencia con respecto a las demás aplicaciones descritas en este punto es que no permite comunicación textual, ya que, para ello, Apple tiene disponible otra aplicación, iMessage.

En enero de 2019, fue descubierto un bug en la aplicación, mediante el cual se podía entablar una llamada con una persona, aunque la llamada no fuera aceptada. Por tanto, se podía escuchar el audio del usuario receptor, aunque éste no acepte la llamada. El bug se solucionó en un breve periodo de tiempo [11].

### **2.2.4 Discord**

Es una herramienta que, en principio, está diseñada para usuarios que juegan a videojuegos, pero se puede usar perfectamente como una aplicación de comunicación como las que se han descrito anteriormente. Es multiplataforma, está disponible tanto como para macOS, iOS, Android, Windows y Linux. Además, posee una web mediante la cual se puede disfrutar de su funcionalidad.

Para usar la aplicación no es necesario ningún tipo de login, sin embargo, es necesario tener una cuenta de correo electrónico para realizar el proceso de verificación del usuario.

En cada sesión, se pueden abrir servidores (conversaciones) en las que se pueden invitar a otros usuarios y mantener conversaciones con ellos mediante voz, audio o vídeo. Al ser una aplicación diseñada para ser usada mientras los usuarios juegan a videojuegos, tiene la propiedad de tener una latencia muy baja.



## **3 CAPÍTULO 3 – ANÁLISIS DE LA SOLUCIÓN**

En este capítulo se realiza un estudio de las diferentes soluciones que se podrían implementar para realizar el proyecto, escogiendo de manera justificada una de ellas, que será la que se implemente. Después, se realiza una comparativa entre las aplicaciones disponibles en el mercado, explicadas en el punto anterior con la solución escogida para la realización de este trabajo.

### **3.1 Selección de la solución**

Para conseguir los objetivos del proyecto se pueden tomar diversos caminos, no hay una solución única. En este caso, hay varios aspectos a tener en cuenta a la hora de elegir la solución:

1. Tipo de aplicación a implementar.
2. Lenguaje de programación a usar.
3. Tecnología a usar para comunicar los dispositivos.
4. Protocolo a usar para comunicar los dispositivos.

#### **3.1.1 Elección del tipo de aplicación a implementar**

A la hora de realizar una aplicación para dispositivos móviles, se tiene que elegir una de las tres formas que hay disponibles:

1. Aplicación nativa.
2. Aplicación web.
3. Aplicación híbrida.

##### *3.1.1.1 Aplicación nativa*

Este tipo de aplicaciones son las que están implementadas con el kit de desarrollo nativo de cada sistema operativo. De esta manera, si se quiere hacer una aplicación que funcione

en dos sistemas operativos, como Android e iOS, hay que implementar dos versiones: una para Android y otra para iOS.

Este tipo de aplicaciones tiene como principal ventaja el poder acceder a cualquiera de las funcionalidades del dispositivo en el que está ejecutando (siempre que el usuario de permiso) como es la cámara, el calendario, el gps, etc. Otra de sus ventajas es que pueden funcionar, aunque el dispositivo no tenga conexión a internet.

El principal inconveniente que tienen es el coste, principalmente por dos razones: el coste de programación suele ser más alto y, además, si se quiere hacer que una aplicación funcione en varios sistemas operativos, hay que hacer una versión por sistema operativo. El tiempo de desarrollo, por tanto, es más elevado.

### *3.1.1.2 Aplicación web*

Este tipo de aplicaciones son páginas web que se visualizan desde el navegador web disponible en un dispositivo, presentándose la funcionalidad en el cliente mediante la interacción con un servidor web. Por tanto, para programarla se usa HTML, CSS, JavaScript etc.

La principal ventaja de este tipo de aplicaciones es que puede funcionar en cualquier tipo de móvil, no es necesario tener una versión por sistema operativo. Esto hace que su coste sea muy bajo.

Tiene, sin embargo, dos inconvenientes:

- Para poder funcionar, necesitan siempre conexión a internet.
- No tiene acceso a las funcionalidades del dispositivo, como la cámara, gps etc, ya que la aplicación es un simple visor de contenido. Esto puede traer problemas ya que, por ejemplo, hace que no se pueden enviar notificaciones al usuario.

Usar este tipo de aplicaciones es recomendable (por su bajo coste y tiempo de trabajo) si se tiene disponible una página web y se quiere implementar una aplicación móvil para visualizar el contenido de esa web.



### 3.1.1.3 Aplicación híbrida

Las aplicaciones híbridas están basadas en una página web, de la que se muestra su contenido y una serie de mejoras que se añade en la aplicación para que esta pueda acceder a las funcionalidades del sistema operativo. Los frameworks más habituales con los que se desarrollan estas aplicaciones son Electron y Apache Cordova.

Tiene varias ventajas, como la multiplataforma: no es necesario tener una aplicación diferente por cada sistema operativo. Esta ventaja tiene otra asociada: el coste. El tiempo dedicado a la programación es mucho más bajo, ya que se necesita una sola aplicación para los diferentes sistemas operativos y parte del contenido de la aplicación se basa en una página web.

En cuanto a la experiencia del usuario, el rendimiento suele ser más bajo que el de una aplicación nativa, pero, por otro lado, puede ofrecer funcionalidades más complejas que una aplicación web e incluso funcionar sin conexión a Internet.

Una vez se conocen los tres tipos de aplicaciones existentes, es necesario realizar un pequeño análisis para ver cuál es la que más se ajusta a este proyecto:

1. Uno de los objetivos a conseguir es que la aplicación funcione sin conexión a internet. Es necesario que la aplicación sea nativa o híbrida.
2. Se necesita que la aplicación acceda al micrófono y altavoces del dispositivo que se está usando y a funciones avanzadas de comunicación (Wi-Fi, Bluetooth). Una aplicación web queda descartada, quedando como alternativa una aplicación nativa o híbrida.
3. En este proyecto no se tiene ninguna página web previa, y desarrollar la aplicación utilizando tecnologías web podría resultar más complejo. Por tanto, una aplicación web o híbrida queda descartada, siendo necesaria una aplicación nativa.

Por tanto, el tipo de aplicación que se va a desarrollar es una **aplicación nativa**.

### **3.1.2 Elección del lenguaje de programación**

Al implementarse una aplicación nativa, tenemos tres lenguajes de programación candidatos a usar: C++, Java o Kotlin.

C++ queda descartado en un principio porque la dificultad de programación en Android es alta. Sólo se recurrirá a este lenguaje si durante el desarrollo del proyecto se considera que la latencia obtenida en algunos de los puntos (captura del audio del micrófono, envío de audio) es demasiado elevada para lo requerido en este trabajo.

Ahora ya tenemos solo dos opciones disponibles: Java o Kotlin. Nos podemos basar en la elección en si los lenguajes tienen disponible las funciones de Android y el SDK y aquí, los dos lenguajes trabajan de la misma manera: ambos tienen disponibles las bibliotecas y SDK de Android y ejecutan en la máquina virtual de Java de los dispositivos. No podemos utilizar este criterio para elegir entre uno y otro, por lo que se usan otros criterios para realizar la elección.

El ingeniero que ha realizado este proyecto, conoce Java ya que es el primer lenguaje que se enseña en la universidad. Si se decide usar Kotlin, hay que estimar un alto tiempo de aprendizaje, ya que un lenguaje de programación no se aprende en un mes, y más para un proyecto de esta envergadura. En este aspecto, Java tiene un punto a favor. Además, al ser Kotlin un lenguaje bastante nuevo, aún no hay en internet tanta documentación y tutoriales como hay disponible para Java ni la comunidad de usuarios es tan alta, por lo que buscar información si se necesita en algún momento del proyecto, será más complicado. Otro punto a favor para Java.

El código en Kotlin suele ser más corto que en Java ya que se necesitan menos líneas para realizar una tarea. Esta es una ventaja, ya que depurar el código será más sencillo, pero para una persona sin experiencia, puede ser más sencillo usar Java ya que se ve más claramente (al haber más líneas) lo que se ha programado.

Por estas razones, se elige **Java** como lenguaje de programación: es un lenguaje que conoce la persona que realiza este proyecto y se salva así tener que aprender un nuevo lenguaje que, a priori no da unas ventajas muy significativas frente a Java. Además, el objetivo de este proyecto no es aprender un nuevo lenguaje de programación, por lo que Java es perfectamente válido.

### **3.1.3 Elección de la tecnología a usar para comunicar los dispositivos**

Hay disponibles dos tecnologías para comunicar los dispositivos: Bluetooth y Wi-Fi Direct.

Al estar plateándose una aplicación de comunicación de audio, se requiere que los envíos tengan poca latencia para que la conversación sea posible de una manera natural. Bluetooth tiene el problema de que suele tener una latencia alta y que, aunque en algunas versiones la latencia ha disminuido, no todos los dispositivos la soportan.

Para solucionar este problema de latencia, se podría plantear la opción de usarlo simplemente para la mediación a la hora de conectar dispositivos a la misma red Wi-Fi (por ejemplo, compartiendo las credenciales de acceso), y que sea a través de Wi-Fi como se realice después el envío del audio, ya que este tipo de redes experimentan baja latencia.

Sin embargo, si se utiliza Wi-Fi Direct este proceso de coordinación es automático y no hace falta que exista un punto de acceso a una red Wi-Fi si no que la conexión es directa, lo que ahorra pasos de configuración (usando mediación por Bluetooth probablemente se necesite intervención del usuario para conectarse a la red Wi-Fi) y permite depender de una única tecnología.

Por tanto, se elige **Wi-Fi Direct** para como tecnología para comunicar a los dispositivos.

### 3.1.4 Elección del protocolo a usar para comunicar los dispositivos

Aunque la opción ideal sería utilizar un protocolo P2P para que los dispositivos tengan una carga equilibrada (cada dispositivo haría broadcast del audio de su micrófono y recibiría el audio del resto de dispositivos, que mezclaría y reproduciría por su altavoz), esto no es posible en la tecnología Wi-Fi Direct ya que está orientada a que un dispositivo del grupo actúe como servidor.

Por lo tanto, si se elige como tecnología de comunicación Wi-Fi Direct, es inevitable utilizar el esquema **cliente-servidor** a menos que se dedique una gran cantidad de tiempo a diseñar una solución software que haga que la aplicación funcione como P2P.

### 3.2 Comparativa de la solución con trabajos similares

En este apartado se realiza una pequeña comparativa entre las aplicaciones disponibles en el mercado explicadas en el apartado anterior (Skype, Hangouts, Facetime y Discord) con la implementada en este proyecto.

TABLA 3.1: COMPARATIVA ENTRE ALGUNAS APLICACIONES DISPONIBLES EN EL MERCADO Y LA IMPLEMENTADA EN ESTE PROYECTO

	Skype	Hangouts	Facetime	Discord	Aplicación implementada
Conexión a internet necesaria para poder funcionar	SÍ	SÍ	SÍ	SÍ	NO
Funciona en más de una plataforma	SÍ	SÍ	SÍ	SÍ	NO
Servidor necesario para realizar la comunicación	SÍ	SÍ	SÍ	SÍ	NO
Distancia mínima necesaria entre los dispositivos para poder funcionar	NO	NO	NO	NO	SÍ
Requieren de un proceso de registro para poder usarse	SÍ	SÍ	SÍ	SÍ	NO

- Las aplicaciones con las que se comparara este proyecto necesitan conexión a internet, la implementada no. Esto puede ser tomado como una ventaja o inconveniente, dependiendo del uso que se desee dar a la aplicación. Con conexión a internet salvamos la distancia existente entre los dispositivos. Con la aplicación de este proyecto, hay una distancia mínima entre los dispositivos, pero se puede llevar a cabo el proceso de comunicación en sitios donde la conexión a internet sea nula como en sótanos, aviones, etc. Además, al no necesitar conexión a internet la aplicación es inmune a una caída puntual de la conexión a internet.
- El resto de aplicaciones son multiplataforma. La desarrollada en este proyecto solo funciona en Android, por lo que está en desventaja con respecto a las que hay ya disponibles en el mercado.
- Las aplicaciones con las que se ha comparado el proyecto, necesitan un servidor entre los dispositivos para realizar el proceso de comunicación. La realizada en este proyecto, no. Atendiendo al uso que se desee dar a la aplicación puede ser una ventaja o una desventaja: con un servidor, se necesita conexión a internet, pero no es necesario que los dispositivos estén cerca. Sin el servidor, no es necesaria la conexión a internet, pero los dispositivos deben estar próximos entre sí. Además, la ausencia de un servidor entre los dispositivos, hace que la aplicación implementada no tenga problemas de funcionamiento por una hipotética caída del servidor (ya que no existe) y evita que un cierto país bloquee su uso, ya que al no tener ninguna conexión a internet es imposible hacerlo. A nivel de privacidad también es una ventaja, ya que la conversación entre los dispositivos va de emisor a receptor, evitando que terceras personas puedan realizar algún tipo de espionaje gracias al servidor.

La ausencia de servidor, tiene como desventaja la falta de un registro de las conversaciones que el usuario ha tenido con otro usuario. Sin este registro, no se puede guardar información de con quien se ha comunicado un usuario, la duración de las llamadas, etc. A nivel de privacidad es una ventaja, pero una desventaja si el usuario necesita un histórico de uso.

- Como ya se ha comentado en este apartado, la aplicación implementada necesita una distancia mínima entre los dispositivos para poder funcionar. El resto de aplicaciones, no.

- La aplicación de este proyecto no requiere ningún proceso de login o registro para poder funcionar. Esto es una ventaja ya que un usuario no necesita ceder datos a terceros ni gastar tiempo en realizar un login o algún registro para poder usar la aplicación.

## **4 CAPÍTULO 4 – DISEÑO E IMPLEMENTACIÓN DEL SISTEMA**

En este capítulo se realiza una explicación del diseño e implementación de la aplicación que se va a implementar. La aplicación pasa por distintas fases hasta que el proceso de comunicación comience a funcionar. Al estar basada en Wi-Fi Direct, estas fases están muy ligadas a los distintos procesos que tienen lugar en el estándar.

### **4.1 Fases**

A continuación, se realiza una explicación de cada una de las fases.

#### **4.1.1 Fase de análisis de compatibilidad y estado**

Aunque la mayoría de dispositivos móviles que ejecutan Android en la actualidad soportan el estándar Wi-Fi Direct a través de una implementación determinada, sigue existiendo dependencia del controlador de Wi-Fi que proporcione el fabricante del dispositivo. La aplicación realizará un primer proceso de inicialización de servicios de Wi-Fi Direct en el que podrá comprobar si el dispositivo es apto o no para la comunicación. En caso de que, por cualquier motivo, no se pueda acceder a las funcionalidades de Wi-Fi Direct, se muestra un mensaje de error para informar al usuario de que el dispositivo no puede ejecutar la aplicación.

El caso del simulador de Android Studio es especial. Aunque no produce ningún error por compatibilidad como el que aparecería en un dispositivo móvil, la mayoría de funciones disponibles para Wi-Fi Direct muestran una situación de inactividad. Es posible que esto se deba al proceso de simulación, que aísla el entorno Android de la configuración de red del dispositivo que se encarga de la simulación.

Para implementar el análisis de compatibilidad y controlar el estado de la aplicación a lo largo de su ejecución, se recurren a las funcionalidades proporcionadas por Android SDK. El entorno de desarrollo de Android permite capturar ciertos eventos del sistema a través de uno o más manejadores. En el caso de Wi-Fi Direct, los eventos que interesa capturar se encuentran en la clase WifiP2pManager y son:

- ***WIFI\_P2P\_STATE\_CHANGED\_ACTION***: el estado de Wi-Fi Direct ha cambiado. La causa más habitual suele ser la activación o desactivación de la conexión Wi-Fi del dispositivo, aunque también puede darse lugar durante las conexiones o desconexiones de un punto de acceso a Internet.
- ***WIFI\_P2P\_PEERS\_CHANGED\_ACTION***: la lista de dispositivos disponibles ha cambiado. Esto sucede cada vez que un dispositivo deja de estar disponible o cuando aparecen nuevos dispositivos al alcance de Wi-Fi Direct.
- ***WIFI\_P2P\_CONNECTION\_CHANGED\_ACTION***: la conectividad de Wi-Fi Direct ha cambiado. Este evento se produce cuando se ha conectado o desconectado el dispositivo por Wi-Fi Direct con otro dispositivo.
- ***WIFI\_P2P\_THIS\_DEVICE\_CHANGED\_ACTION***: los detalles de configuración de Wi-Fi Direct han cambiado. Este cambio de configuración puede requerir acciones adicionales por parte de la aplicación, por lo que debe atenderse.

Una vez registrada la captura de estos eventos en el dispositivo a través de la función de Android *registerReceiver*, se debe obtener una referencia al subsistema de Wi-Fi Direct mediante la siguiente función:

*(WifiP2pManager) getSystemService(Context.WIFI\_P2P\_SERVICE)*

La inicialización del servicio desencadenará el lanzamiento del manejador con cada evento capturado y nos proporcionará un canal (*WifiP2pManager.Channel*) con el que podremos referenciar la instancia inicializada. A partir de ese momento, se considerará que la aplicación ya puede realizar sus funciones, ya que, mientras tanto, no hay demasiadas tareas para realizar.

Cualquier error a la hora de ejecutar funciones con *WifiP2pManager* que se asocie con *P2P\_UNSUPPORTED* significará que el dispositivo no es compatible con Wi-Fi Direct.

#### **4.1.2 Fase de exploración**

En caso de que se considere que el dispositivo es compatible con Wi-Fi Direct, se puede continuar con la siguiente fase; la fase de exploración. En ella, el dispositivo *informa* a



otros dispositivos cercanos de su disponibilidad para iniciar una sesión de comunicación. Al mismo tiempo, *recopila* un listado de otros dispositivos que se encuentren informando de esta misma condición en su entorno. Este procedimiento es similar al que tendría lugar en otras tecnologías, como Bluetooth.

La distancia a la que tiene que encontrarse otro dispositivo para que pueda detectarse depende tanto de la sensibilidad del módulo de red del dispositivo móvil que realiza la detección como del alcance del que está informando de su disponibilidad. Por lo general, la experiencia ha mostrado que es una distancia similar a la necesaria para conectar con un punto de acceso Wi-Fi en el mismo entorno.

En la siguiente figura, se describe cómo tendría lugar una fase de exploración en la que participan 4 dispositivos que se encuentran a poca distancia.

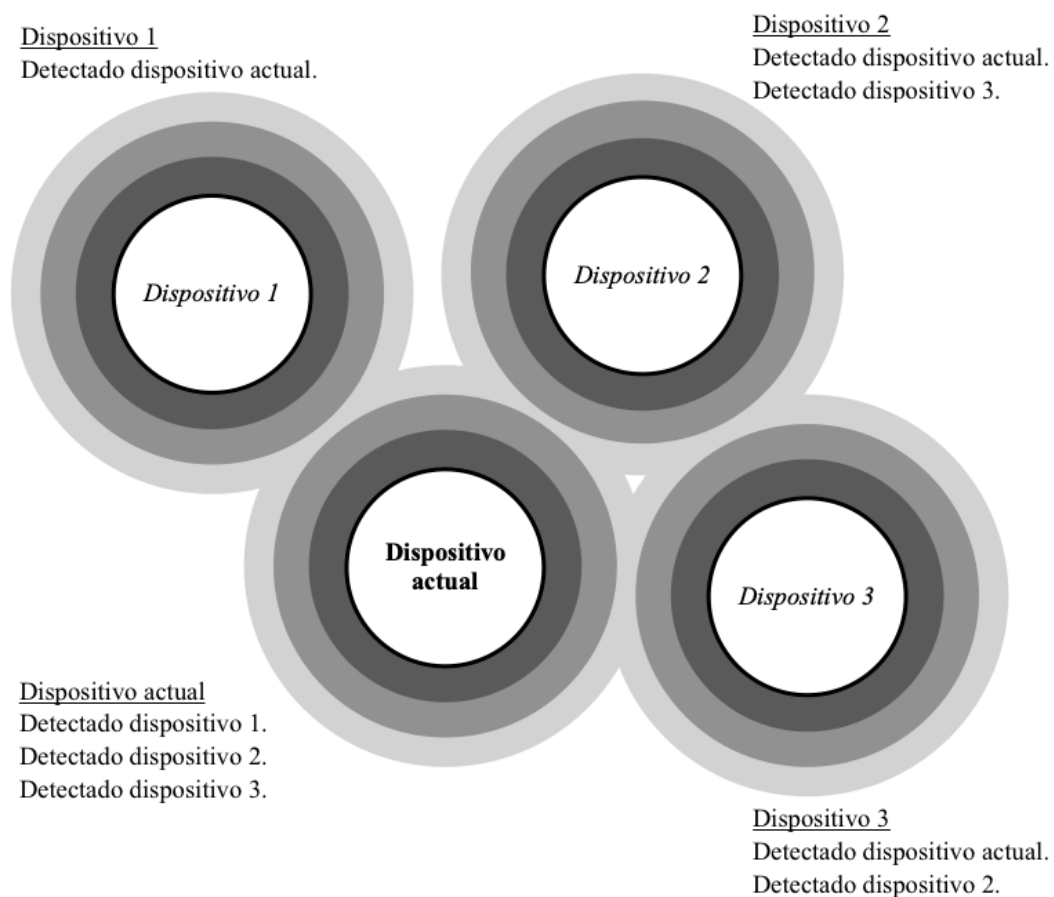


Fig 4.1: Descripción de la fase de exploración

Es importante señalar que, en la implementación actual, no se descarta ningún dispositivo disponible en la red, aunque éste no esté ejecutando la aplicación definida en este trabajo. En el apartado 8.2 *Líneas de trabajo futuras*, se indica la manera en que podría mejorarse esta situación para sólo contemplar dispositivos que estén en la fase de exploración y ejecutando esta aplicación.

Por último, en la *Fig 2: Pantalla de exploración de los dispositivos disponibles*, se presenta la pantalla de exploración de dispositivos de la aplicación, que permite visualizar todos los dispositivos encontrados al realizar la exploración Wi-Fi. Dependiendo de la localización del usuario, la lista puede tener mayor o menor número de dispositivos. En lugares muy concurridos, como aeropuertos u hoteles, la lista puede ser muy larga. Mediante los gestos de la interfaz, se puede hacer scroll de la lista para visualizar todos los dispositivos disponibles.

La lista, además, se actualiza en tiempo real. Esto significa que, en cuanto un dispositivo pase a estar fuera del alcance de Wi-Fi, desaparecerá de la lista. Lo mismo pasa si un dispositivo pasa a estar dentro del alcance de red inalámbrica.

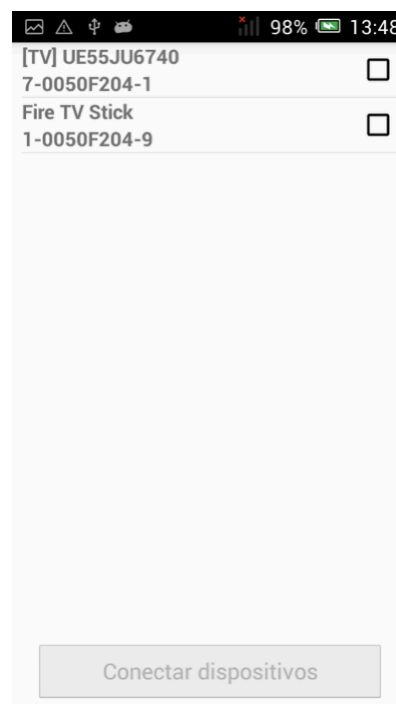


Fig 4.2: Pantalla de exploración de los dispositivos disponibles

Para recuperar estos dispositivos desde el SDK de Android, se recurre a la instancia de *WifiP2pManager*. La función *discoverPeers* permite, indicando el canal, iniciar el descubrimiento de dispositivos y, al mismo tiempo, exponer la identidad del dispositivo para que otros puedan conectarse.

Una vez iniciado el descubrimiento, se puede invocar a la función *requestPeers* de la instancia de *WifiP2pManager* para obtener la lista de dispositivos descubiertos. Esto debe hacerse cada vez que se produzca un evento del tipo *WIFI\_P2P\_PEERS\_CHANGED\_ACTION* para mantener la lista siempre actualizada.

#### **4.1.3 Fase de conexión**

Una vez se ha expuesto de manera gráfica al usuario el listado con los dispositivos detectados en sus cercanías, se le permite seleccionar los dispositivos con los que desea iniciar la comunicación. En este punto, se pone en marcha el mecanismo de conexión de Wi-Fi Direct.

Cabe destacar que, aunque se ha dejado abierta la posibilidad de seleccionar múltiples dispositivos para la conferencia, sólo se soporta, por el momento, la conexión con un dispositivo. De nuevo, esta situación se detalla en el apartado 8.2 *Líneas de trabajo futuras*.



Fig 4.3: Selección de un dispositivo para la conexión

En anterior figura, se puede apreciar la selección de un dispositivo con el que realizar la conexión. La pantalla sigue siendo la misma que para la exploración de dispositivos disponibles (Fig 4.2 *Pantalla de exploración de los dispositivos disponibles*). El botón para conectar dispositivos sólo se encuentra activo cuando un dispositivo ha sido seleccionado, ya que, de otro modo, no tendría sentido su pulsación.

Cuando el usuario pulsa el botón de conectar con algún elemento de la lista seleccionado, se invoca la función *connect* de la instancia de *WifiP2pManager*. Al igual que para iniciar el descubrimiento de dispositivos, es necesario indicar el canal. Esta vez, además, se debe suministrar una configuración *WifiP2pConfig*, en la que se indica la dirección del dispositivo con el que se quiere conectar y se incorpora el tipo de seguridad Wi-Fi Protected Setup que mantendrá la conexión. Para evitar que el usuario deba introducir un código en el dispositivo para iniciar la conexión, se utiliza el sistema *PBC* (*Push Button Configuration*).

Si la conexión entre los dispositivos ha tenido éxito, se recibirá un evento de tipo *WIFI\_P2P\_CONNECTION\_CHANGED\_ACTION*. Cuando se esté procesando este evento, se dispondrá de la información de la conexión invocando la función

*requestConnectionInfo* de la instancia de *WifiP2pManager*. Hay dos datos que son de especial importancia para los siguientes pasos a realizar:

- ***isGroupOwner***: Indica si el dispositivo actual es el propietario del grupo o no. El propietario del grupo en Wi-Fi Direct es el dispositivo móvil que actúa con el rol de servidor dentro del grupo. El resto de dispositivos tienen que pasar necesariamente por él para comunicarse con otros dentro del grupo. Para elegir este dispositivo, Wi-Fi Direct examina las características de los dispositivos que van a formar el grupo y elige el que sea más adecuado para actuar con este rol [12].
- ***groupOwnerAddress***: Indica la dirección asignada al propietario del grupo. Esta información es útil para todos los dispositivos del grupo independientemente de su rol. En el caso del propietario del grupo, servirá para saber dónde puede exponer sus servicios. Para el resto de dispositivos, es necesaria para saber dónde deben contactar para consumir los servicios.

Una vez disponible esta información, se considerará que la conexión ha tenido éxito y se transitará a la siguiente fase.

#### **4.1.4 Fase de comunicación**

Cuando los dispositivos se encuentran conectados, tendrán distinta operativa dependiendo del rol que se les haya asignado en Wi-Fi Direct. Las tareas comunes que tienen los dispositivos son:

- *Capturar* audio a través de su micrófono y *enviarlo* al otro dispositivo.
- *Recibir* audio de otro dispositivo y *reproducirlo* a través del altavoz o los auriculares.

El esquema planteado es simple cuando se tienen dos dispositivos, ya que el propietario del grupo tendrá un funcionamiento muy similar al de cualquier otro dispositivo del grupo, como se puede apreciar en la siguiente figura.



Fig 4.4: Comunicación entre dos dispositivos

Para que la comunicación tenga la menor latencia posible pero, a la vez, conseguir que la información de audio llegue ordenada y sin pérdidas sustanciales de información, se utilizan sockets TCP para realizar el envío y recepción de datos. El propietario del grupo abre un socket que vincula a su dirección asignada para Wi-Fi Direct y a un puerto conocido y elegido a priori para la aplicación, el 8888, y permanece escuchando peticiones de conexión.

Por otro lado, el dispositivo no propietario del grupo abre un socket vinculado a un puerto aleatorio y lo conecta al socket del propietario del grupo, cuya dirección ya conoce al haberse recibido al crear la conexión y cuyo puerto también conoce porque es siempre el mismo para la aplicación.

Una vez conectados los sockets, el propietario del grupo mandará a través del socket del cliente el flujo de audio que captura a través de su micrófono y recibirá a través de este mismo socket el audio capturado por el otro dispositivo del grupo, reproduciéndolo después por su altavoz. El dispositivo que no es propietario del grupo hará exactamente lo mismo a través de su socket de cliente.

Si el esquema planteado fuera para N dispositivos, el funcionamiento sería más complejo, como se puede apreciar en la siguiente figura:

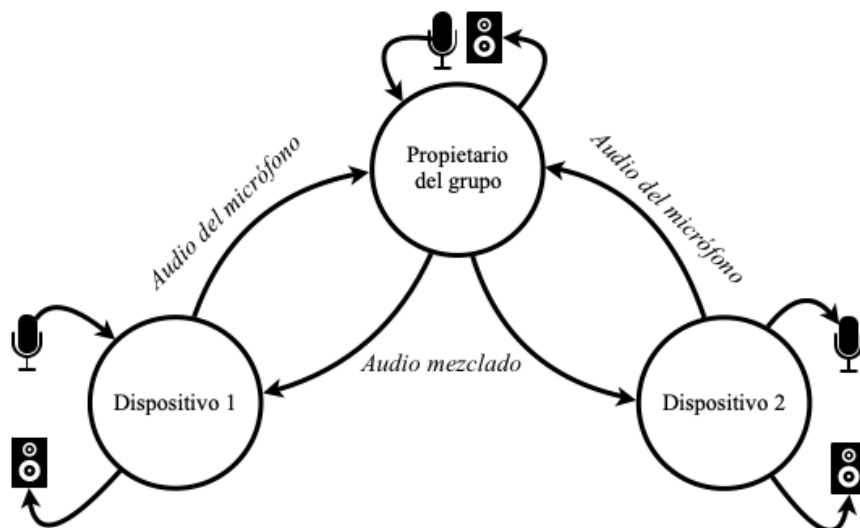


Fig 4.5: Comunicación entre N dispositivos

Este esquema requiere funcionalidad adicional para el propietario del grupo. Su funcionalidad pasa a ser más crítica que la del resto de dispositivos, ya que recibe  $N-1$  señales de audio procedentes del micrófono de los demás dispositivos. El propietario deberá mezclar todas estas señales en una sola, ya que los altavoces o auriculares sólo aceptan un único flujo de audio para reproducir.

Sin embargo, la parte compleja viene a la hora de enviar la mezcla a los demás dispositivos. No basta con mezclar todas las entradas de micrófono del resto de dispositivos con la entrada de micrófono del propietario del grupo y enviarlas a todos los dispositivos, ya que, entonces, el usuario de un dispositivo concreto estaría recibiendo por el altavoz su propio audio del micrófono. En su lugar, habría que enviar una mezcla distinta a cada uno de los dispositivos conectados, en la que no se incorpore su señal de micrófono. Este proceso podría plantear retos para mantener la latencia en presencia de un grupo grande de dispositivos, ya que el proceso de mezcla requeriría cierta capacidad de procesamiento y tiene que suceder en tiempo real.

Al ser una tarea mucho más compleja que cuando se tienen únicamente dos dispositivos, se ha implementado la opción planteada en *Fig 4.4 Comunicación entre dos dispositivos*. Sin embargo, se describe la manera en que podría realizarse la comunicación entre  $N$

dispositivos por si se quisiera continuar este trabajo y expandirlo a este caso más avanzado.

A nivel de interfaz de usuario, se utiliza una pantalla con diferente aspecto en función del rol del dispositivo: el fondo es rojo si el dispositivo ha sido designado propietario del grupo, y verde si el dispositivo no ha adquirido este rol, como se puede observar la siguiente figura:

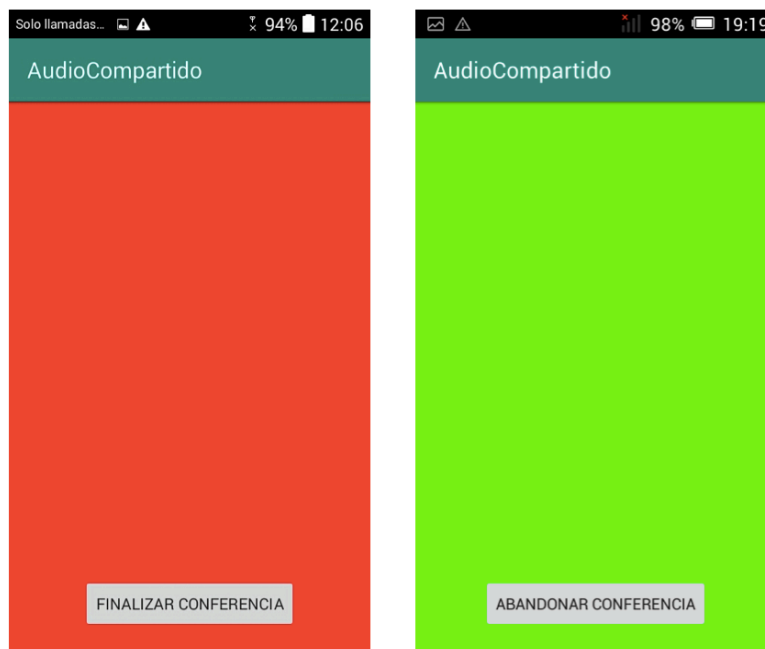


Fig 4.6: Pantalla de conferencia



Mientras el usuario no decida abandonar la conferencia, el audio de su micrófono seguirá siendo enviado y se seguirá reproduciendo el procedente del otro dispositivo.

#### **4.1.5 Fase de desconexión**

En la última fase, la fase de desconexión, el dispositivo deja de estar vinculado a la conferencia de audio y retorna a la fase de exploración. Esta fase se desencadena al pulsar el botón “Finalizar conferencia” o “Abandonar conferencia” de la pantalla de conferencia (*Fig 4.6 Pantalla de conferencia*), aunque también puede alcanzarse por motivos inesperados: desplazar el dispositivo fuera del alcance de la cobertura de Wi-Fi o apagar el Wi-Fi desde la configuración de Android.

Existe un matiz a la hora de realizar la desconexión. Si el dispositivo que abandona es el propietario del grupo, el botón indica “Finalizar conferencia” porque, en caso de abandonar, todos los demás dispositivos también lo harán como consecuencia al haber perdido el dispositivo que actúa como servidor. En la implementación actual, tendrán que crear una nueva conferencia para que se asigne un nuevo propietario de grupo. Si, por el contrario, abandona el grupo otro dispositivo pero el propietario sigue vinculado, entonces la conferencia podrá seguir adelante sin ese dispositivo. Como en la implementación actual sólo se contempla que participen dos dispositivos en la conferencia, esta cuestión tampoco supone un problema porque no tiene sentido una conferencia en la que sólo participa un dispositivo.

Para conseguir que las conferencias puedan mantenerse incluso si el propietario del grupo las abandona, habría que asegurar una “transferencia de poderes” en el momento de su desconexión. Otro dispositivo debería adquirir su rol y el resto de dispositivos deberían mandar sus peticiones a la nueva dirección. Esta funcionalidad también resulta compleja, por lo que se plantea en este documento como trabajo futuro.

En Android SDK, la desconexión del grupo se consigue invocando la función *removeGroup* en la instancia de *WifiP2pManager*. Cuando la desconexión se haya producido, se recibirá un evento del tipo *WIFI\_P2P\_CONNECTION\_CHANGED\_ACTION*. Es en este momento cuando se finaliza la actividad Android vinculada a la conferencia, regresándose a la pantalla de

exploración de dispositivos disponibles (Fig 2: *Pantalla de exploración de dispositivos móviles*).

## 4.2 Resumen de funcionamiento

Una vez exploradas las fases, se procede a resumir el funcionamiento de la aplicación. La aplicación consta de dos actividades Android y cuatro estados. En el grafo presentado en la siguiente figura, se representan todas las fases de la aplicación y las condiciones que llevan a un cambio de fase.

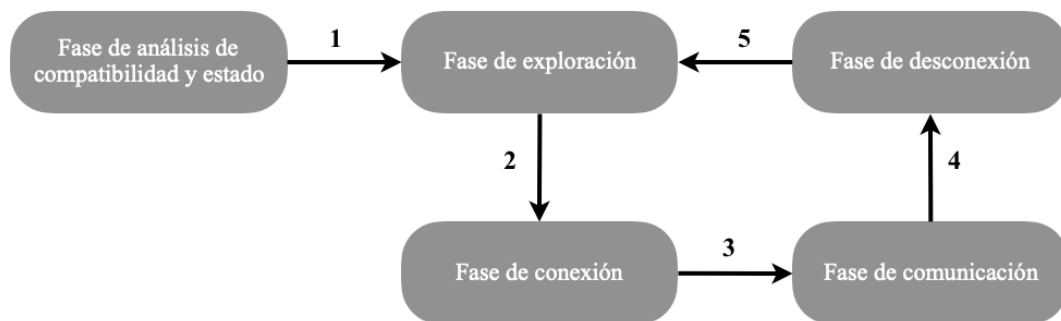


Fig 4.7: Grafo de transición entre fases

- **Transición 1:** el dispositivo es compatible con Wi-Fi Direct y su controlador Wi-Fi permite su utilización para este tipo de redes.
- **Transición 2:** el usuario ha seleccionado un dispositivo móvil de la lista con el que conectar.
- **Transición 3:** la conexión tiene éxito y se ha creado un grupo de conexión entre los dispositivos.
- **Transición 4:** el usuario ha pulsado sobre el botón para finalizar o abandonar la conferencia, o se ha perdido la conexión con los dispositivos por desactivación del Wi-Fi o pérdida de cobertura.
- **Transición 5:** la desconexión tiene éxito.

### 4.3 Captura de sonido

Para la captura de audio del micrófono, se utiliza la clase *AudioRecord* disponible en Android SDK. El formato utilizado para la grabación es el soportado por la mayoría de dispositivos Android disponibles: 16.000 muestras por segundo, un sólo canal, codificado como PCM de 16 bits. Este formato debe ser el mismo en todos los dispositivos para reducir la latencia que implicaría realizar conversiones de formato. Además, debe ser el mismo que se utilice para la reproducción de audio.

Para que el audio procedente del micrófono se reciba en los dispositivos a la mayor velocidad posible, se debe utilizar un tamaño de buffer relativamente bajo para el micrófono. Sin embargo, no todos los dispositivos soportan tamaños de buffer pequeños. Al igual que con el formato, se ha elegido un tamaño que dota de compatibilidad con la mayoría de dispositivos: 8.192 muestras.

### 4.4 Reproducción del audio

Para la reproducción de audio en los auriculares o altavoces, se utiliza la clase *AudioTrack* disponible en Android SDK. La instancia utilizada se configura con el mismo formato que el que se utiliza para la captura de audio en todos los dispositivos. El modo de suministro de datos se coloca en *MODE\_STREAM*, ya que las muestras se van a ir proporcionando en tiempo real. Por otro lado, se utiliza el tipo de stream *STREAM\_VOICE\_CALL*, que hace que Android trate el flujo de audio de la aplicación como si se tratara de una llamada de teléfono, permitiendo que se respeten el volumen de reproducción y otros parámetros configurados por el usuario para el audio de las llamadas.



## **5 CAPÍTULO 5 – PLANIFICACIÓN**

A continuación, se expone la planificación inicial de este proyecto y se realiza un análisis de dicha planificación para ver si se ha sido la correcta y se ha cumplido durante el tiempo de duración del proyecto.

### **5.1 Planificación inicial**

El proyecto se dividió inicialmente en una serie de sprints. Cada sprint tiene como objetivo ejecutar las siguientes tareas:

1. SPRINT 1: Una vez definidos los objetivos del proyecto, realización de la planificación del proyecto.
2. SPRINT 2: Realización de estudio de alternativas de solución al problema planteado. Realización de la documentación asociada al estudio de alternativas de solución.
3. SPRINT 3: Diseño de la solución, una vez se haya seleccionado con qué alternativa se soluciona el problema planteado.
4. SPRINT 4: Implementación del diseño escogido. Familiarización y estudio con las tecnologías escogidas para solucionar el problema. Instalación del entorno de trabajo.
5. SPRINT 5: Realización de pruebas y comprobación de que se cumple la funcionalidad esperada, solucionándose el problema planteado.
6. SPRINT 6: Realización del resto de documentación necesaria.

A continuación, se adjunta un diagrama de Gantt donde se muestran los hitos expuestos, así como su duración a lo largo del tiempo:

TABLA 5.1: DIAGRAMA DE GANTT DE LA PLANIFICACIÓN INICIAL DEL PROYECTO

Sprint	Mes 1				Mes 2				Mes 3				Mes 4			
Sprint 1																
Sprint 2																
Sprint 3																
Sprint 4																
Sprint 5																
Sprint 6																

Como se puede observar en la tabla anterior, la duración estimada del proyecto es de 4 meses, siendo la duración de estimada (en semanas) de cada sprint la siguiente:

TABLA 5.2: DURACIÓN ESTIMADA DE CADA SPRINT EN SEMANAS

Sprint	Duración estimada (semanas)
Sprint 1	1 semana
Sprint 2	2 semanas
Sprint 3	2 semanas
Sprint 4	6 semanas
Sprint 5	2 semanas
Sprint 6	3 semanas
<b>TOTAL</b>	<b>16 semanas</b>

## 5.2 Duración real del proyecto

Como la planificación inicial del proyecto es de 4 meses, la fecha de comienzo fue el 11 de febrero, cuatro meses antes de la fecha límite de entrega de la convocatoria de junio. Sin embargo, la planificación no se ha seguido por las causas descritas a continuación:

- En el SPRINT 3, se seleccionó una solución al problema que no era la más adecuada, ya que se decidió realizar la aplicación basada en Bluetooth en lugar de en WiFi-Direct. A la hora de la implementación, se observó que la latencia era elevada. Por tanto, se tuvo que realizar de nuevo el SPRINT 3 y SPRINT 4.
- El SPRINT 6 se alargó una semana más de lo esperado, ya que se realizó una mala estimación del tiempo de realización de la documentación asociada al proyecto.

La duración real de cada sprint ha sido la siguiente:

TABLA 5.3: DURACIÓN REAL DE CADA SPRINT EN SEMANAS

<b>Sprint</b>	<b>Duración real (semanas)</b>
Sprint 1	1 semana
Sprint 2	2 semanas
Sprint 3*	4 semanas
Sprint 4*	12 semanas
Sprint 5	2 semanas
Sprint 6*	4 semanas
<b>TOTAL</b>	<b>25 semanas</b>

Los sprints marcados con \* son los que han sufrido una duración más elevada a la que se planificó. Esto ha hecho que todo el proyecto se retrase, ya que de las 16 semanas de debía durar, ha durado 25 semanas (alrededor de 6 meses). Es por esto que el proyecto se ha presentado en la convocatoria de septiembre.





## 6 CAPÍTULO 6 – ENTORNO SOCIO-ECONÓMICO

En este apartado, en primer lugar, se realiza un presupuesto del trabajo completo. También se detalla el impacto socio-económico que tendría el proyecto puesto en producción una vez haya sido ejecutado.

### 6.1 Presupuesto

En este apartado se realiza una estimación del coste real del proyecto, dividiéndose de la siguiente manera:

- En primer lugar, se detallan los gastos directos asociados al proyecto.
- En segundo lugar, se exponen los gastos indirectos y el porcentaje de beneficio asociado al proyecto.
- Por último, se calcula el presupuesto total, teniendo en cuenta tanto los gastos directos como los indirectos.

El presupuesto se da en euros (€)

#### 6.1.1 Gastos directos

Los gastos directos asociados a un proyecto de desarrollo de software son los siguientes:

1. Coste del personal que trabaja en el proyecto.
2. Coste del software.
3. Coste del hardware.

El total de los gastos directos es de 20.094,15 euros. A continuación, se detalla cada uno de ellos:

##### *6.1.1.1 Coste del personal que trabaja en el proyecto.*

El proyecto ha sido realizado por un ingeniero informático. El tiempo de duración real del proyecto ha sido de 25 semanas, trabajando en el proyecto de lunes a viernes durante

4 horas al día. Una semana, tiene 5 días laborables, por lo que, haciendo un cálculo sencillo se sabe que las horas totales invertidas en el proyecto han sido 500.

Se toma que el salario de un ingeniero informático junior es de 40 euros por hora. Por tanto, el coste total en concepto de salario es de 20.000 euros (€).

En el coste personal también habría que añadir dietas y transporte. El coste en concepto de dietas y transporte es de 0 euros (€) ya que el proyecto ha sido realizado en casa del ingeniero informático, no siendo necesario ningún tipo de desplazamiento. Al ser 4 horas el tiempo diario dedicado al proyecto, no aplica cargar al proyecto ningún coste en dietas.

TABLA 6.1: COSTE DEL PERSONAL QUE TRABAJA EN EL PROYECTO

Concepto	Coste (€)
Salario	20.000 €
Transporte	0 €
Dietas	0 €
<b>TOTAL</b>	<b>20.000 €</b>

Por tanto, el coste total del personal que trabaja en el proyecto es de 20.000 euros.

#### *6.1.1.2 Coste del software*

Durante el periodo de duración del proyecto, se han usado diferentes softwares y programas informáticos. Algunos de ellos son gratuitos, otros tienen un coste. A continuación, se presenta el desglose por el coste del software.

Los programas informáticos necesarios para realizar el proyecto han sido los siguientes:

- El entorno de desarrollo integrado utilizado ha sido Android Studio. La licencia que posee permite su uso de manera gratuita, por lo que el costo es de 0 euros.

- En la implementación del código, se han usado las bibliotecas que están presentes en el Android SDK. Se permite su uso de manera gratuita, por lo que el costo en licencias de uso de bibliotecas es de 0 euros.
- Para realizar la documentación del proyecto (el presente documento), se ha usado el programa informático Microsoft Word. La licencia obtenida es la de Office 365 Personal, que tiene un precio de 69 euros al año. Este precio no es el que se debe cargar al proyecto, se debe cargar únicamente el coste de la amortización. Para calcular la amortización, la Agencia Tributaria Española define unas tablas [13] en las que especifica el coeficiente lineal máximo y el periodo máximo de vida útil que tiene un determinado elemento.

Para calcular el coste de amortización de un producto, se toma su precio inicial y se resta al valor residual del producto<sup>1</sup> una vez haya pasado su vida útil. El resultado de esa operación se multiplica por el coeficiente lineal máximo, que da como resultado el coste de amortización del producto en un año. Ese dato, se divide entre los días que tiene un año, teniendo así el coste de amortización diario. Se multiplica ese valor por los días que ha durado el proyecto<sup>2</sup>, teniendo como resultado final el coste de amortización del producto que se puede cargar al proyecto.

TABLA 6.2: COSTE DEL SOFTWARE

Concepto	Precio inicial (€)	Valor residual (€)	Coeficiente lineal máximo	Coste de amortización (€)
Android Studio	0 €	0 €	33%	0 €
Android SDK	0 €	0 €	33%	0 €
Office 365 Personal	69 €	20,70 €	33%	5,46 €
<b>TOTAL</b>				<b>5,46 €</b>

En resumen, el coste total del software utilizado en el proyecto ha sido de 5,46 euros.

<sup>1</sup> El valor residual de un producto es un dato que se supone, en este proyecto se toma como el 30% del precio inicial del producto.

<sup>2</sup> El proyecto ha durado 25 semanas. Si hay 5 días laborables a la semana, el proyecto ha durado 125 días.

### 6.1.1.3 Coste de los aparatos electrónicos e informáticos

Durante el tiempo de realización del proyecto, se han usado una serie de equipos electrónicos e informáticos, por lo tanto, se carga al proyecto su coste de amortización.

TABLA 6.3: COSTE DE LOS APARATOS ELECTRÓNICOS E INFORMÁTICOS

Concepto	Precio inicial (€)	Valor residual (€)	Coefficiente lineal máximo	Coste de amortización (€)
Teclado y ratón inalámbrico	0 €	0 €	20%	0 €
Monitor BENQ	150 €	45 €	20%	7,19 €
Ordenador portátil (MacBook Air)	1400 €	420 €	20%	67,12 €
Alcatel OneTouch	120 €	36 €	20%	5,75 €
Huawei Y625	180 €	54 €	20%	8,63 €
<b>TOTAL</b>				<b>88,69 €</b>

Por tanto, el total del coste de los aparatos electrónicos e informáticos asciende a 88,69 euros.

### 6.1.2 Gastos indirectos y porcentaje de beneficio

Los costes indirectos son aquellos que no se pueden cargar directamente al proyecto pero que son necesarios para que éste se pueda realizar. Los gastos indirectos asociados a este proyecto son los que se presentan en la siguiente tabla:

TABLA 6.4: COSTES INDIRECTOS ASOCIADOS AL PROYECTO

Concepto	Justificación
Factura de la luz y agua	Para la realización del proyecto, es necesario tener electricidad para alimentar los dispositivos electrónicos y agua en el espacio de trabajo
Alquiler del lugar de trabajo	El espacio de trabajo es una vivienda de alquiler
Factura de conexión a internet	Para la realización del proyecto, es necesario tener conexión a internet para buscar documentación, acceder al repositorio donde se guarda el código, etc

Al ser la propia vivienda del ingeniero el lugar de desarrollo del trabajo, es difícil saber la cantidad exacta de los costes indirectos. Por tanto, se realiza una aproximación y, se cobrará un 15% del total de los costes directos en concepto de costes indirectos. El total de los costes directos es de 20.094,15 euros. Si se calcula el 15% de esa cifra, se obtiene que el total de costes indirectos es de 3.014,12 euros.

También, es necesario obtener un porcentaje de beneficio por la realización del proyecto. Se toma como porcentaje de beneficio un 15% del valor de los costes directos. Por tanto, se obtiene que el coste que se debe cargar al proyecto en concepto de beneficios es de 3.014,12 euros.

El coste total en concepto de costes indirectos y de beneficios es de 6.028,25 euros.

#### 6.1.2.1 Coste total

El coste total del proyecto (sin IVA) es la suma de los costes directos, coste indirectos y beneficio.

TABLA 6.5: COSTE TOTAL DEL PROYECTO (SIN IVA)

Concepto	Coste (€)
Costes directos	20.094,15 €
Costes indirectos	3.014,12 €
Beneficio	3.014,12 €
<b>TOTAL</b>	<b>26.122,39 €</b>

En los proyectos de desarrollo de software, el porcentaje de IVA a aplicar es del 21%. EL IVA queda por tanto en 5485,70 €.

**El coste total del proyecto (IVA incluido) es de 31608,09 €**

## **6.2 Impacto socio-económico**

Para analizar el impacto que tendrá la aplicación en la sociedad y en el ámbito económico, hay que analizar, en primer lugar, la situación en que se encuentran en la actualidad.

### **6.2.1 Análisis del panorama actual**

A nivel tecnológico, la sociedad está experimentando los avances en las tecnologías de la telecomunicación. Las redes móviles actuales tienen gran cobertura 4G [14] [15] [16], por lo que hay amplia disponibilidad de conexión a Internet de alta velocidad, y se está comenzando a implantar la tecnología 5G en España [17], que plantea mejoras del ancho de banda. En cuanto a Wi-Fi, se están empezando a vender dispositivos que soportan el nuevo estándar Wi-Fi 6 [18], que mejora el alcance, ancho de banda y consumo eléctrico frente a los estándares anteriores. Por último, la tecnología Bluetooth presenta un nuevo estándar, Bluetooth 5 [19], que disminuye la latencia en la transmisión de datos, y que empezó a implantarse en dispositivos entorno al año 2017 [20].

La oferta de tarifas mensuales de datos móviles y fibra óptica de los operadores móviles también han disminuido drásticamente de precio, impulsadas por los operadores virtuales y de bajo coste [21]. De esta manera, se reducen los recursos necesarios para tener acceso a una conexión de alta velocidad y con cuotas de datos amplias.

Por otra parte, la amplia oferta en el mercado de dispositivos móviles está permitiendo la proliferación de alternativas de coste muy reducido y de características cada vez más notables [22].

Por lo tanto, a nivel social, se está ampliando el acceso a las redes a un público cada vez más amplio, al estar reduciéndose la barrera económica: cada vez es más asequible tener un dispositivo móvil conectado a Internet.

### **6.2.2 Análisis del impacto social**

Aunque el panorama actual indica una tendencia hacia el acceso universal a Internet, existen zonas rurales y espacios urbanos en los que existe mala o nula cobertura de datos. Además, existen restricciones en las redes corporativas que impiden que cualquier usuario pueda conectarse a la red. Es decir, siguen existiendo espacios donde las circunstancias físicas o las condiciones de seguridad plantean una barrera para el acceso a Internet.

Es en estos espacios donde la aplicación planteada en este proyecto tendrá un impacto notorio, ya que las demás alternativas existentes no funcionarán ante la falta de conexión a Internet o la cobertura baja.

En países de menor desarrollo que España, donde las cuotas de los operadores móviles sean más elevadas y la infraestructura no cubra tanto territorio, puede que una aplicación de estas características sea la única alternativa posible para la comunicación entre personas.

Con el panorama de mejora en las tecnologías de comunicación, es de esperar que el impacto social de esta aplicación se vaya reduciendo paulatinamente a lo largo del tiempo, dada la omnipresencia de la conexión a Internet de alta velocidad. El único aspecto en que puede seguir siendo más accesible es el de la configuración: no es necesario introducir credenciales de ninguna red Wi-Fi, ni tener una cuenta de usuario propia para empezar a utilizar la aplicación. Además, al descartar el uso de servidores, seguirá funcionando ante caídas de infraestructura.

### **6.2.3 Análisis del impacto económico**

De comercializarse una aplicación como la planteada en este trabajo, contribuiría al enriquecimiento del actual mercado de aplicaciones móviles para el sistema operativo Android.

Cabe destacar que la forma de explotar económicamente la aplicación puede ser complicada. La inexistencia de servidores intermedios impide la medición del uso del servicio y hace complicado justificar ante los usuarios el pago de una cuota mensual, ya que no hay gastos en infraestructura. Por lo tanto, la aplicación debería venderse a través de un pago único. Sin embargo, existiendo alternativas gratuitas que funcionan a través de Internet, y entendiendo que esta aplicación no debería plantear barreras económicas para acceder a sus funciones –ya que rompería una de las premisas que causan impacto social– el precio que puede alcanzar es limitado.

Es posible, sin embargo, que se pueda explotar a través de espacios de publicidad. En el diseño e implementación de la aplicación se indicaba que la pantalla de conferencia tiene únicamente un botón para abandonar la misma. Todo el espacio restante podría utilizarse para mostrar información de los participantes y, además, reservar un espacio para la publicidad. Sin embargo, y como consecuencia de que la aplicación esté planteada para funcionar sin conexión a Internet, habría un límite en cuanto a la capacidad de explotación mediante publicidad: los anuncios serían fijos y sería difícil contabilizar el impacto que ha tenido en los usuarios.

Por lo tanto, aunque la propuesta puede aumentar el mercado de aplicaciones móviles y crear mayor competencia en el ámbito de las aplicaciones de comunicación por audio, tiene fuertes restricciones en cuanto al beneficio económico que se puede extraer de su comercialización. Además, al igual que sucedía al analizar el impacto social, la constante mejora en las tecnologías de la telecomunicación reducirá las ventajas de esta aplicación a lo largo del tiempo más allá de la de la configuración inmediata.



## **7 CAPÍTULO 7 – MARCO REGULADOR**

En este capítulo se realiza un análisis de la legislación aplicable al proyecto.

### **7.1 Privacidad y protección de datos**

La aplicación descrita a lo largo de este trabajo gestiona un medio de entrada de especial sensibilidad, como es el micrófono. Por este motivo, hay que vigilar con especial atención el cumplimiento de la última normativa disponible en España en materia de protección de datos.

En España, la ley que regula esta cuestión en la actualidad es la Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y Garantía de los Derechos Digitales (LOPD-GDD) [23], que sustituyó a la Ley Orgánica 15/1999 de Protección de Datos de Carácter Personal (LOPD) [24]. Esta ley se redactó en consonancia con el Reglamento General de Protección de Datos (RGPD) [25] de la Unión Europea, que a su vez entró en vigor el 25 de mayo de 2018.

Las personas tienen derecho, bajo el amparo de esta ley, al acceso, rectificación, supresión, oposición, limitación del tratamiento y portabilidad de sus datos personales [23]. Cualquier omisión de estos derechos o fallo a la hora de proteger los datos personales de carácter sensible implica sanciones acordes a la gravedad de la infracción [23].

Un análisis minucioso de la aplicación revela dos momentos en los que la información debe ser protegida: procesado y transporte del audio.

### **7.2 Procesado del audio**

La aplicación no almacena en el almacenamiento persistente del dispositivo móvil los datos relativos al envío o recepción de audio durante la comunicación, pero sí que puede encolar en memoria el audio entrante –en caso de que aún no pueda ser procesado por el sistema de reproducción de audio de Android–, o el audio saliente, –en caso de que la conexión con el dispositivo de destino no sea igual de rápida que la obtención de muestras del micrófono–. Android proporciona aislamiento de la memoria de las aplicaciones para

impedir que aplicaciones maliciosas puedan obtener acceso a información sensible de otras aplicaciones, por lo que no es necesario, a priori, mantener cifrados los buffers de audio descritos.

### **7.3 Transporte del audio**

Durante la fase de conexión entre dispositivos, Wi-Fi Direct debe informar a otros dispositivos de la disponibilidad para la conexión. El identificador que se utiliza en los dispositivos empleados durante las pruebas se compone de datos sobre el dispositivo móvil, como el nombre del fabricante, el modelo o la generación, aunque el estándar no especifica la manera exacta en la que se deben componer dichos identificadores [26]. Esta información permite que otros usuarios de la aplicación puedan conectar con el dispositivo adecuado. El identificador no permite asociar el dispositivo móvil con una persona concreta, por lo que no sirve por sí mismo para determinar información personal de un individuo.

El envío de muestras de audio, una vez confirmada la conexión entre dispositivos, tiene lugar a través de una red Wi-Fi privada entre ambos, por lo que puede considerarse de punto a punto. Sin embargo, esto no es suficiente para garantizar la confidencialidad de la comunicación, ya que todo intercambio realizado a través de un canal público –en este caso, el aire– es susceptible de ser interceptado, analizado y comprendido por un tercero. El estándar de Wi-Fi Direct plantea distintas formas de cifrado, como Wi-Fi Protected Setup y WPA, para las comunicaciones. Sin embargo, la forma concreta en que se cifran las comunicaciones vendrá determinada por la implementación del estándar, siendo las implementaciones de los estándares más antiguos menos restrictivas en cuanto al cifrado [26]. Lo más habitual en Android es que el mecanismo sea el planteado por Wi-Fi Protected Setup [27]. Aunque podría aplicarse un cifrado de los datos de audio a nivel de aplicación, salvaguardando las comunicaciones con independencia de la implementación de Wi-Fi Direct, esto ha probado incrementar la latencia en la recepción y el envío de las muestras, dificultando la comunicación entre los usuarios. Por este motivo, y entendiendo este proceso de cifrado manteniendo la latencia como un reto, se plantea el cifrado a nivel de aplicación como trabajo futuro.

### **7.4 Propiedad intelectual**

Tanto el código fuente de la aplicación móvil como esta memoria escrita son creaciones originales. Como cualquier obra original, por lo tanto, debe elegirse la forma en que quedarán protegidos los derechos de autor.

### **7.5 Memoria escrita**

El presente documento se encuentra sujeto a la licencia Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional [28]. Esta licencia no permite adaptaciones del documento ni tampoco su uso con fines comerciales, pero sí acepta la publicación y transferencia de la obra siempre y cuando se mantenga la atribución de autoría.

### **7.6 Código fuente**

El código fuente vinculado a la aplicación móvil, excluyendo sus dependencias de terceros, se proporciona bajo licencia GNU General Public License v2.0 [29]. Esta licencia concede, a grandes rasgos, los siguientes derechos:

- Copiar y distribuir el código fuente de la aplicación sin modificar.
- Modificar el código fuente de la aplicación y distribuir la versión modificada.
- Distribuir versiones compiladas de la aplicación (por ejemplo, el APK), modificadas o no, siempre y cuando:
  - Todas las copias distribuidas vengan acompañadas de una copia de la licencia y la exclusión de cualquier garantía.
  - Todas las copias se distribuyen bajo la misma licencia GPLv2.
  - Todas las versiones compiladas de la aplicación vienen acompañadas del código fuente correspondiente, u ofrecen una manera viable de acceder al mismo.

### **7.7 Dependencias**

El código fuente de la aplicación no utiliza bibliotecas adicionales ni herramientas de terceros, salvo las contenidas en el kit de desarrollo de software de Android (Android SDK).

Las cláusulas de la licencia que otorga Google para utilizar Android SDK [30] permiten el desarrollo de aplicaciones exclusivamente para Android. También obligan a la no

eliminación, disimulo ni alteración de posibles avisos de derechos de propiedad intelectual que pudieran anexarse o incluirse en el SDK.

Por último, la propiedad de Android SDK sigue siendo de sus legítimos propietarios y no se está proporcionando el SDK como parte de la aplicación, ni en forma compilada ni como código fuente (el dispositivo móvil es el que, posteriormente, contiene estos elementos); por lo que no se vulnera ningún acuerdo de licencia ni ninguna protección de propiedad intelectual, y no es necesario añadir información de licencia adicional más allá de la propia del software. Se entiende, de este modo, que no hay ningún conflicto legal a la hora de utilizar este entorno para el desarrollo.

## 8 CAPÍTULO 8 – CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

En este capítulo se exponen las conclusiones una vez ha finalizado el proyecto. Además, se añaden una serie de propuestas que se pueden realizar en el futuro para mejorar la aplicación implementada

### 8.1 Conclusiones

La aplicación desarrollada ha seguido las premisas planteadas: funciona para móviles con sistema operativo Android, funciona si hay dos dispositivos conectados y permite comunicar, compartiendo audio de manera bidireccional, a las personas que usen esos dispositivos, aunque no exista conexión a internet. Por tanto, se puede considerar que el proyecto ha sido un éxito en cuanto al cumplimiento de los objetivos.

El hecho de que no se haya conseguido seguir (por las causas explicadas en el capítulo 5.2 *Duración real del proyecto*) la planificación inicial puede considerarse un fracaso. Ha habido un retraso de aproximadamente dos meses con respecto a lo que se planificó. Por tanto, este aspecto debe ser un punto de mejora para futuros proyectos, realizar mejores planificaciones.

Otras de las conclusiones que se arrojan de este proyecto es que Wi-Fi Direct es muy restrictivo a la hora de trabajar con otro protocolo de comunicación que no sea cliente-servidor, es necesario mucho trabajo de desarrollo si se desea trabajar con otro protocolo.

Gracias al desarrollo de este proyecto, he podido aplicar a un proyecto muchos de los conocimientos adquiridos durante mi etapa de estudiante del Grado en Ingeniería Informática en la Universidad Carlos III de Madrid. Además, me ha servido para aprender, ya que he mejorado mi conocimiento tanto a nivel teórico como práctico en Wi-Fi Direct, Bluetooth y Android.

## 8.2 Líneas de trabajo futuras

Hay varios aspectos en los que se puede seguir mejorando y trabajando para hacer una aplicación más atractiva:

- Hacer la aplicación multiplataforma, haciendo una versión para iOS.
- Filtrar la lista de dispositivos encontrados por Wi-Fi Direct para que solo aparezcan los dispositivos que usan la aplicación plantrada. Para ello, se puede utilizar el concepto de servicio que se plantea en el estándar de Wi-Fi Direct y que permite que un dispositivo informe sobre el servicio que está exponiendo o al que se quiere conectar al resto de dispositivos que se encuentran en exploración [31].
- Permitir la conexión mediante Wi-Fi Direct entre  $n$  dispositivos. Aunque esta funcionalidad está implementada en el código, la comunicación posterior entre dispositivos no lo está, por lo que quedaría pendiente verificar que esta conexión entre  $n$  dispositivos funciona correctamente.
- Permitir la comunicación mediante Wi-Fi Direct entre  $n$  dispositivos. En el *Capítulo 4 - Diseño e implementación* del sistema se describen los retos que aparecen para conseguirlo:
  - Crear mezclas de audio específicas para cada dispositivo.
  - Mantener unos niveles de latencia aceptables incluso cuando esta mezcla se realiza en tiempo real.
  - Conseguir que el proceso de mezcla sea lo suficientemente ligero como para que el dispositivo que actúa como servidor pueda llevarlo a cabo incluso si sus características fueran modestas.
- Cifrar las comunicaciones a nivel de aplicación para mantener la seguridad incluso si la implementación de Wi-Fi Direct del dispositivo utilizase un mecanismo de cifrado de Wi-Fi inseguro. La principal dificultad en este caso es mantener una baja latencia.

## BIBLIOGRAFÍA

- [1] R. v. d. Meulen, «Gartner,» [En línea]. Available: <https://www.gartner.com/en/newsroom/press-releases/2018-02-22-gartner-says-worldwide-sales-of-smartphones-recorded-first-ever-decline-during-the-fourth-quarter-of-2017>.
- [2] R. v. d. Meulen, «Gartner,» [En línea]. Available: <https://www.gartner.com/en/newsroom/press-releases/2018-02-22-gartner-says-worldwide-sales-of-smartphones-recorded-first-ever-decline-during-the-fourth-quarter-of-2017>.
- [3] O. H. Alliance, 2007 noviembre 2008. [En línea]. Available: [http://www.openhandsetalliance.com/press\\_110507.html](http://www.openhandsetalliance.com/press_110507.html).
- [4] C. Collado, «andro4all,» [En línea]. Available: <https://andro4all.com/2019/08/android-10-fecha-lanzamiento>.
- [5] J. Titus, «Android Developers,» [En línea]. Available: <https://android-developers.googleblog.com/2017/05/google-io-2017-empowering-developers-to.html>.
- [6] W.-F. Alliance, «Wi-Fi Alliance,» [En línea]. Available: <https://www.wi-fi.org/knowledge-center/faq/how-far-does-a-wi-fi-direct-connection-travel>.
- [7] W.-F. Alliance, «Wi-Fi Alliance,» [En línea]. Available: <https://www.wi-fi.org/knowledge-center/faq/how-does-security-work-for-wi-fi-direct-certified-products>.
- [8] A. R. Castellano, «Bluetooth. Introducción a su Funcionamiento.,» Universidad Pontificia de Comillas, Madrid, 2011.
- [9] C. M. Pascual, «Bluetooth: criterios de selección y comparativa con otras tecnologías inalámbricas,» Tecnica Industrial, 2012.
- [10] Skype, «Skype,» [En línea]. Available: <https://support.skype.com/es/faq/fa31/usa-skype-cifrado>.
- [11] E. Perez, «Xataka,» [En línea]. Available: <https://www.xataka.com/privacidad/grave-fallo-facetime-permite-escuchar-remotamente-otros-iphone-antes-que-respondan-llamada>.

- [12] D. Android, «Developer Android,» [En línea]. Available: <https://developer.android.com/training/connect-devices-wirelessly/wifi-direct>.
- [13] A. T. Española, «Agencia Tributaria,» [En línea]. Available: [https://www.agenciatributaria.es/AEAT.internet/Inicio/\\_Segmentos\\_/Empresas\\_y\\_profesionales/Empresas/Impuesto\\_sobre\\_Sociedades/Periodos\\_impositivos\\_a\\_partir\\_de\\_1\\_1\\_2015/Base\\_imponible/Amortizacion/Tabla\\_de\\_coeficientes\\_de\\_amortizacion\\_lineal\\_.shtml](https://www.agenciatributaria.es/AEAT.internet/Inicio/_Segmentos_/Empresas_y_profesionales/Empresas/Impuesto_sobre_Sociedades/Periodos_impositivos_a_partir_de_1_1_2015/Base_imponible/Amortizacion/Tabla_de_coeficientes_de_amortizacion_lineal_.shtml).
- [14] Movistar, «movistar,» [En línea]. Available: <http://www.movistar.es/particulares/coberturas/movil/4G/>.
- [15] Orange, «Orange,» [En línea]. Available: <https://4g.orange.es/>.
- [16] V. España, «Vodafone,» [En línea]. Available: <https://www.vodafone.es/c/conocenos/es/vodafone-espana/mapa-cobertura-movil/>.
- [17] R. Muñoz, «El País,» [En línea]. Available: [https://elpais.com/economia/2019/06/10/actualidad/1560156504\\_005782.html](https://elpais.com/economia/2019/06/10/actualidad/1560156504_005782.html).
- [18] R. Limón, «El País,» [En línea]. Available: [https://elpais.com/tecnologia/2019/09/16/actualidad/1568653377\\_406392.html](https://elpais.com/tecnologia/2019/09/16/actualidad/1568653377_406392.html).
- [19] Europapress, «Europapress,» [En línea]. Available: <https://www.europapress.es/portaltic/sector/noticia-bluetooth-50-principales-ventajas-nueva-tecnologia-inalambrica-smartphones-insignia-20171101112943.html>.
- [20] S. O'Kane, «The Verge,» [En línea]. Available: <https://www.theverge.com/2017/3/29/15112646/samsung-galaxy-s8-bluetooth-5-headphones>.
- [21] R. Muñoz, «El País,» [En línea]. Available: [https://elpais.com/economia/2019/05/03/actualidad/1556897029\\_350170.html](https://elpais.com/economia/2019/05/03/actualidad/1556897029_350170.html).
- [22] J. M. Zuriarrain, «El País,» [En línea]. Available: [https://elpais.com/tecnologia/2017/12/05/actualidad/1512475802\\_638486.html](https://elpais.com/tecnologia/2017/12/05/actualidad/1512475802_638486.html).
- [23] Boletín Oficial del Estado, «Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.,» [En línea]. Available: <https://www.boe.es/eli/es/lo/2018/12/05/3>.



- [24] Boletín Oficial del Estado, «Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal.» [En línea]. Available: <https://www.boe.es/eli/es/lo/1999/12/13/15>.
- [25] Diario Oficial de la Unión Europea, «REGLAMENTO (UE) 2016/679 DEL PARLAMENTO EUROPEO Y DEL CONSEJO,» [En línea]. Available: <https://www.boe.es/doue/2016/119/L00001-00088.pdf>.
- [26] P2P Technical Group, *Wi-Fi Peer-to-Peer (P2P) Technical Specification*, W. Alliance, Ed., 2016.
- [27] Wi-Fi Alliance, *Wi-Fi Protected Setup Specification*, 2006.
- [28] Creative Commons, «Attribution-NonCommercial-NoDerivatives 4.0 International,» [En línea]. Available: <https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>.
- [29] GNU, «GNU General Public License, version 2,» [En línea]. Available: <https://www.gnu.org/licenses/old-licenses/gpl-2.0.html>.
- [30] Google, «Términos y Condiciones de SDK,» [En línea]. Available: <https://developer.android.com/studio/terms>.
- [31] Google, «Use Wi-Fi Direct (P2P) for service discovery,» [En línea]. Available: <https://developer.android.com/training/connect-devices-wirelessly/nsd-wifi-direct>.



# **ANEXO I: RESUMEN EN INGLÉS**

## **AI.1 Abstract**

The goal for this project is to implement a mobile application for Android devices that allows sending audio between devices without an active Internet connection. To reach this goal, a 'conference' (that is, a group of mobile devices) is created. Input audio is recorded through the microphone in each device, then sent real-time to every other device in the conference, and finally played through their default audio output system.

Through a thorough candidate-technology analysis (including Wi-Fi Direct, Bluetooth and P2P networks as communication systems), the most suitable technology found to meet the goal is Wi-Fi Direct. Thus, this is the technology used to implement the application.

## **AI.2 Introduction**

This chapter presents a brief overview of the project. Its area, scope and motivation are detailed, and the goals of this end of degree assignment listed. Finally, the structure of this document is explained.

### **AI.2.1 Overview (Area and Scope)**

This project has been done as an End of Degree Assignment for the Computer Science Degree. Thus, it belongs to the Computer Science area of expertise.

The main goal of this project is to build a Shared Audio System between mobile devices. The system sends audio captured through the microphone of a device to other mobile devices. Therefore, this project requires knowledge on network communication and information theory in order to design and implement an application capable of processing, transmitting and receiving information, presented in an audio form.

Currently, there are two main mobile operating systems: iOS and Android. [1] To meet the goal of this project, the implement application has been developed to support the Android operating system, but not iOS.

The scope of this project also includes the writing of this document, which contains a solution analysis, its technical design and a budget calculation.

### **AI.2.2 Motivation (Context)**

With technological advances, ways of communication have changed radically. There has been a transition from sending written letters and making phone calls to using instant messaging, making video and audio calls through the Internet, etc.

The main purpose of the application implemented in this project is communicating individuals without forcing them to have an active connection to a server on their devices. Connections are created in an end-to-end fashion. This is a useful and fresh approach, as communication can take place anywhere, a mobile device is the only required equipment, nothing else is needed. This changes the traditional way to start communicating, as there is no need to be connected to any data network, have Internet connection or cell coverage. This has a drawback, however: connecting devices directly needs them to be close to each other. If not, devices cannot communicate with each other. It has some advantages too:

- Communication is made possible anywhere (as long as the device the user wants to communicate with is in range): in a boat across the ocean, in a plane, in a basement, etc.
- Privacy is respected. Data does not travel through the Internet, so the chances of data being accessed or stolen are lower.

This is the context in which the application is enclosed.

## AI.3 Goals

The main goal to be achieved through this End of Degree Assignment is to connect two mobile devices so that they can share audio with each other.

The concept of sharing audio is too generic, it can take different forms:

- Sharing a pre-recorded audio track (songs, voice memos, etc.).
- Sharing a real-time audio signal in a bi-directional way (phone call, Skype call, etc.).
- Sharing a real-time audio signal in a unidirectional way (a device acts as a sender and other devices as receivers). The main example of this approach is Walkie-Talkies. While a device is sending audio, it can't receive audio from any other source. Other devices must then wait for its broadcast to end in order to send their own audio.

The goal of this project is to send audio in the second form: sharing a real-time audio signal in a bi-directional way. There are many applications currently that meet this goal, but none of them achieve this without having a server in between, handling device connection and data transfer. Removing the server allows the application to work without a connection to the Internet or to any other shared public network between devices.

This goal may seem too open, so it is split into partial goals:

- The application shall work on the Android operating system.
- The application shall work with two connected devices at least.
- Connection between devices must take place without an Internet connection.

Summing up: the goal is to build a system to share audio between devices in a real-time bi-directional way without requiring a server between them (thus, without an Internet connection).

## **AI.4 Solution analysis**

Once the desired goals are defined, a solution must be selected from a list of candidates for this project.

To achieve the goals for this project, many paths can be taken, not only one. There are some aspects that must be considered when choosing a solution:

1. Type of application to implement.
2. Programming language to use.
3. Technology for communicating devices.
4. Protocol for device communication.

### **AI.4.1 Selection of a type of application to implement**

When developing a mobile application, there are some alternatives as to how to approach its foundations:

1. Native application.
2. Web application.
3. Hybrid application.

A small analysis is performed to decide which option is more convenient for developing the current project:

1. One of the goals of this project is for the application to work without an Internet connection. This requires a native or hybrid application.
2. The application needs access to the microphone, the speaker and some advanced communication features (Wi-Fi, Bluetooth) of the device. A web application is then discarded, only a native or hybrid application can access these subsystems.
3. There is no previous web page that can serve as a base for this project and using web technologies to develop the application would make the task more complex. Thus, a web or hybrid application are not recommended.

After this brief analysis, it is concluded that the developed application will be based on native technologies.

#### **AI.4.2 Selection of a programming language**

When developing native applications for Android, there are three alternatives: Java, Kotlin and C++. Java is the chosen language as it is easier to use, and the developer has previous experience with this language.

#### **AI.4.3 Selection of a technology for communicating devices**

There are two technologies that can enable communication between devices: Bluetooth and Wi-Fi Direct.

The application must send audio to the other device with a little latency in order for the users to be able to communicate without issues. Bluetooth is discarded because not every version of the standard allows for low latency. Wi-Fi Direct is less prone to latency, so it is selected as the communication technology.

#### **AI.4.4 Selection of the protocol for device communications**

Even though the ideal protocol for balanced load between devices would be based on P2P, this is not possible with Wi-Fi Direct as it has a server-client approach at its core: a device has to act as the “group owner” and every device has to go through it to communicate.

Thus, choosing Wi-Fi Direct as the technology for communicating devices entails adopting a client-server approach, unless a great effort is made to develop a software solution that can override this configuration and allow for direct P2P communications.

#### **AI.4.5 Comparison of the selected solution and its alternatives**

In this paragraph, a comparison between four available applications in the market and the implemented solution is made. Functionality on this application is greater than on the developed one, but it is also considered in the table.

**TABLA A.1: COMPARISON BETWEEN SOME AVAILABLE APPLICATIONS ON THE MARKET AND THE SELECTED SOLUTION**

	<b>Skype</b>	<b>Hangouts</b>	<b>Facetime</b>	<b>Discord</b>	<b>Selected solution</b>
Internet connection required	YES	YES	YES	YES	NO
Multi-platform	YES	YES	YES	YES	NO
Server-based communication	YES	YES	YES	YES	NO
Limited range for communication	NO	NO	NO	NO	YES
Sign up required	YES	YES	YES	YES	NO



## AI.5 Design and implementation

The application goes through different stages before the communication process can start. Being based on Wi-Fi Direct, these stages are closely linked to the way this standard works. The stages are defined as follows:

- **Compatibility analysis and status stage:** This stage involves checking device compatibility with Wi-Fi Direct. If found compatible, a series of handlers are set to ensure connection and peer status is captured in the next stages of the application life cycle.
- **Discovery stage:** In this stage, the device informs of its availability to other devices within range, and also compiles a list of devices nearby that support Wi-Fi Direct. This list is then showed to users through the user interface and updated in real time. To identify each device, a friendly name is presented.
- **Connection stage:** When users select a device from the discovered devices list, they have the option to start a conference. Doing this starts a connection process that will create a Wi-Fi Direct group. If the process is successful and the connection is actually created, then the device will move on to the next stage. If not, then it will return to the discovery stage.
- **Communication stage:** When the device has been connected to the Wi-Fi Direct group, a role is assigned to it. This role can either be a “Group owner” or just a regular device in the group. Group owners act as servers in the group, meaning all devices have to communicate with it in order to reach other regular devices. In the two-device scenario supported in this project, the group owner will open a server socket using a known static port. The regular device in the group will open a client socket and connect it to this server socket. This ends network setup, and the devices move on to getting access to their microphone and speaker. Finally, they will send microphone samples through their socket and play the received microphone samples from the other device through their speaker. The user interface in this stage is pretty simple: just a wallpaper and a button that allows to stop communicating with the other device. In a N-to-N device settings, which is not implemented in this project, the process is not that simple. The group owner needs to make a specific mix of microphone inputs for each device in the group,

in order for users not to hear themselves over the speaker. Mixing audio in real time can be a resource consuming task and may increase latency exponentially. This is the reason why the application only implements 1-to-1 communication.

- **Disconnection stage:** Once any of the two users touch the button to stop communicating, or if connection with the other device is lost due to range issues, then the disconnection stage starts its task. The first one: remove the device from the Wi-Fi Direct group. After doing this, the microphone and speaker are returned to their original state and the discovery stage is reached once again, allowing to create new conferences. In a N-to-N device settings, which is not implemented but described for future improvement over this application, disconnection is one of the most challenging stages. Due to Wi-Fi Direct's rigid client-server approach, the disconnection of a group owner will mean the end of the whole conference, as it cannot continue without a server component. For the conference to continue after the group owner disconnects, a complex role reassignment process should take place: some other device in the group should inherit its tasks and every other regular device should reorient its connection towards the new group owner. This is not a simple process, as it also involves handling some network configuration processes manually. This is the reason why this task is left for future works.

#### AI.5.1 Summary of operation

The transitions and stages can be summarized through a simple graph, in order for the reader to understand operations easier.

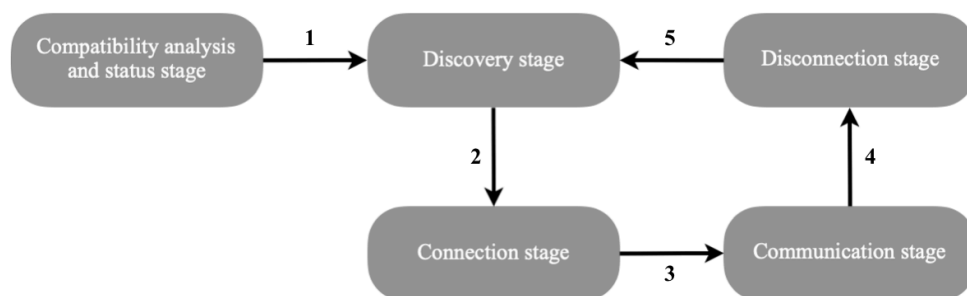


Fig A.1 Stage graph

- **Transition 1:** the device is compatible with Wi-Fi Direct and its Wi-Fi controller allows usage for this kind of networks.
- **Transition 2:** the user has selected a device from the discovered devices list and has pushed the connection button to start communicating with it.
- **Transition 3:** the connection is successful, and a Wi-Fi Direct group has been created between all involved devices.
- **Transition 4:** one of the two users has decided to end the conference, or the connection has been lost due to network issues: one device is now out of range or has disconnected its Wi-Fi.
- **Transition 5:** disconnection has been successful, and the device is now ready to return to discovering Wi-Fi Direct device.

### **AI.5.2 Audio Capture and playback**

Devices capture audio input using their microphones and play audio through their speakers or any connected playback device (like headphones). At a software level, the application uses two Android SDK classes (AudioRecord and AudioTrack respectively) to interact with the underlying hardware devices in a native way.

## **AI.5 Conclusion**

The developed application has followed the defined premises: it works on the Android operating system, it works if two devices are connected, and it allows audio sharing in a bi-directional way, communicating people using this application even if they have no Internet. Therefore, this project has succeeded when it comes to meeting its goals.

The fact that the original schedule was not followed perfectly (because of the reasons detailed in chapter 5.2 *Duración real del proyecto*) can be considered a failure. There was a delay of approximately two months from the original planning. Thus, this aspect has to be improved on future projects: respect the planned schedule.

Another conclusion extracted from this development is that Wi-Fi Direct is restrictive when it comes to the architecture for the application, allowing for little more than a client-server design. Complex software artifacts are required to override this restriction.

Thanks to the development of this project, I could use the knowledge I acquired during my time as a Computer Science student at Universidad Carlos III de Madrid. Moreover, it has given me a chance to learn about some technologies, like Wi-Fi Direct, Bluetooth and Android from a theoretical and practical point of view.

## AI.5 Future lines of work

There are some features that can be improved and worked upon to make the application more attractive:

- Make the application multi-platform, developing a new version for iOS.
- Filter the discovered device list from Wi-Fi Direct so that it only shows devices using this application. This filtering can take advantage of Wi-Fi Direct's concept of "service", that allows devices to inform of the services they provide or consume to other devices in discovery mode that are within range [31].
- Allow Wi-Fi Direct connection between N devices. This functionality is implemented in the source code of the application but is unused as communication afterwards is not yet implemented. It is a pending task to check if this multi-device connection works as expected or not.
- Allow Wi-Fi Direct communication between N devices. *Chapter 4 - Diseño e implementación del sistema* describes the challenges to accomplish this:
  - Create specific audio mixes for each device.
  - Keep a tolerable level of latency while doing this mix in real time.
  - Develop a mixing process that is lightweight to avoid penalizing performance, so that a modest group owner (device chosen as server) can handle mixing all audio in real time.
- Encrypt communication at application level to keep the transmitted data secure even if the Wi-Fi Direct implementation on the device uses weak encryption mechanisms for Wi-Fi transmissions. The main challenge in this case is to keep latency down.